



WHAT REALLY IS pWCET?

A RIGOROUS AXIOMATIC PROPOSAL

RTSS 2023
6 December 2023

Sergey Bozhko, Filip Marković, Georg von der Brüggen, and Björn Brandenburg



MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS



technische universität
dortmund

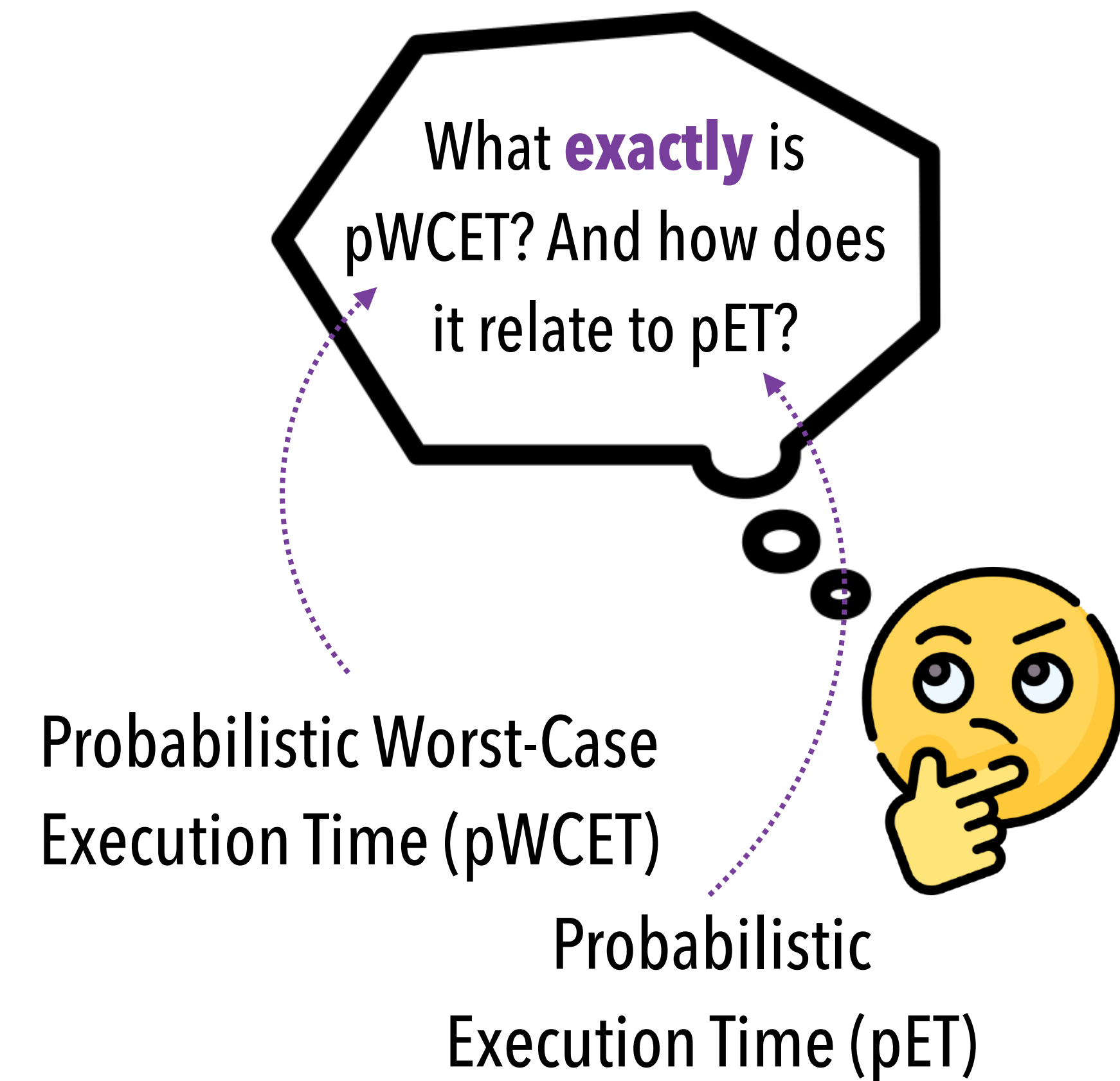


European Research Council
Established by the European Commission



MAX-PLANCK-GESELLSCHAFT

THIS PAPER IN A NUTSHELL



THIS PAPER IN A NUTSHELL

We propose

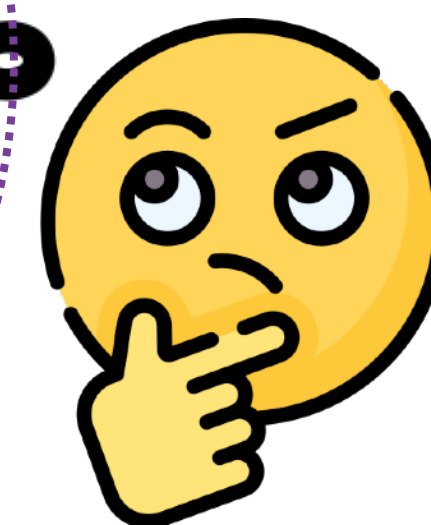
- First **fully formal** definitions of pET and pWCET
- **Adequacy property** capturing the notion of "IID upper bound on pET"
- **Prove** that our proposal of pWCET is adequate in this sense

Independent and identically distributed

What **exactly** is pWCET? And how does it relate to pET?

Probabilistic Worst-Case Execution Time (pWCET)

Probabilistic Execution Time (pET)



THIS PAPER IN A NUTSHELL

We propose

- First **fully formal** definitions of pET and pWCET
- **Adequacy property** capturing the notion of "IID upper bound on pET"
- **Prove** that our proposal of pWCET is adequate in this sense

Independent and identically distributed

We formalized our proposal with the **Coq proof assistant**

- Semantics of stochastic real-time systems
- Definitions of pET, pWCET, and the adequacy property
- **Machine-checked** proof that pWCET is adequate

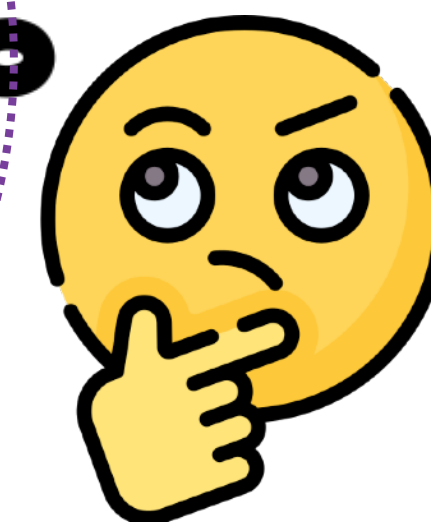


coq.inria.fr

What **exactly** is pWCET? And how does it relate to pET?

Probabilistic Worst-Case Execution Time (pWCET)

Probabilistic Execution Time (pET)



ANALYSIS OF REAL-TIME SYSTEMS: THE BIG PICTURE

THE BIG PICTURE



Predictions

THE BIG PICTURE

Worst-Case Execution
Time (WCET)

To get the predictions right, we need:

- Model with the right **specification**
- *E.g.*, model must include WCETs to allow (classical) response-time analysis

Model of
the system



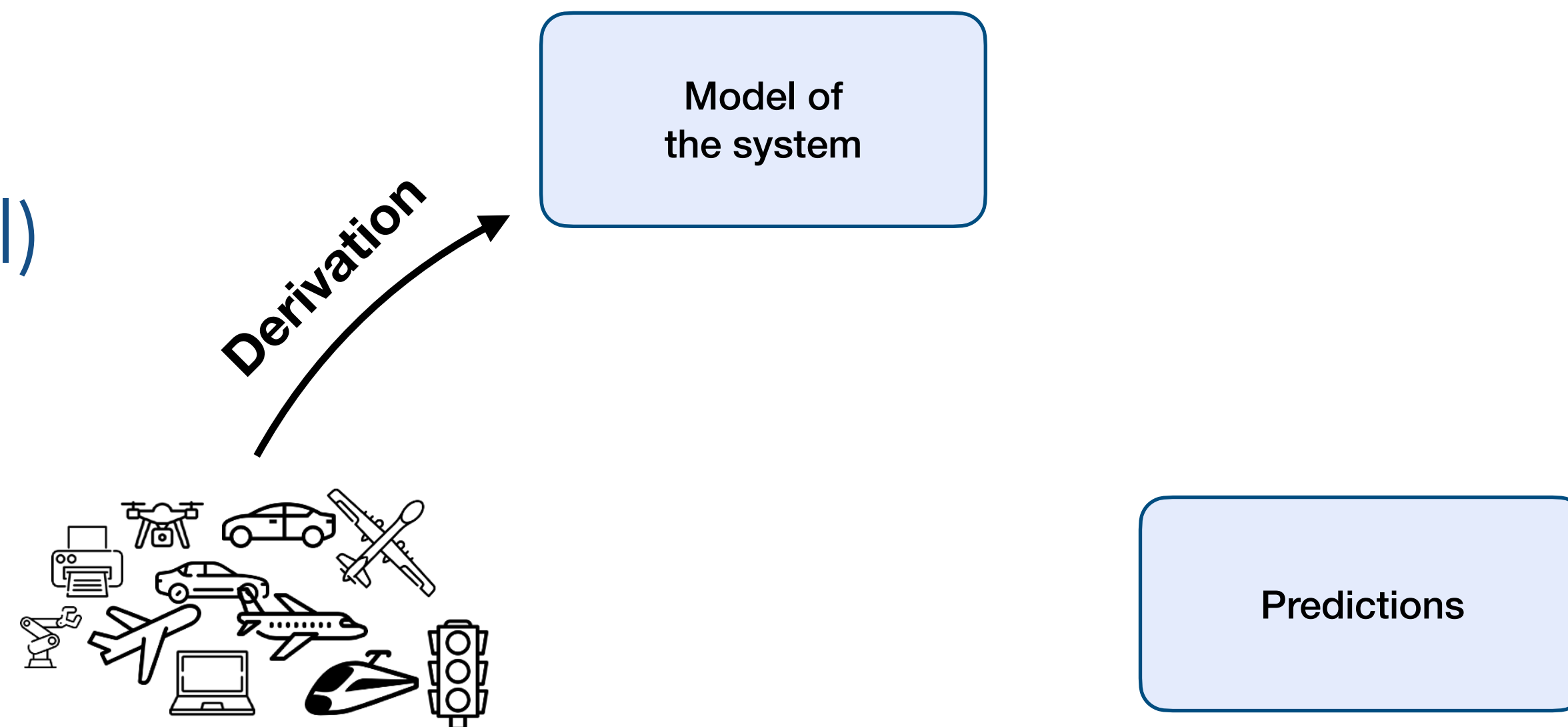
Predictions

THE BIG PICTURE

Worst-Case Execution
Time (WCET)

To get the predictions right, we need:

- Model with the right **specification**
 - *E.g.*, model must include WCETs to allow (classical) response-time analysis
- Correct model **derivation**
 - Optimistic WCET bound \implies Wrong predictions

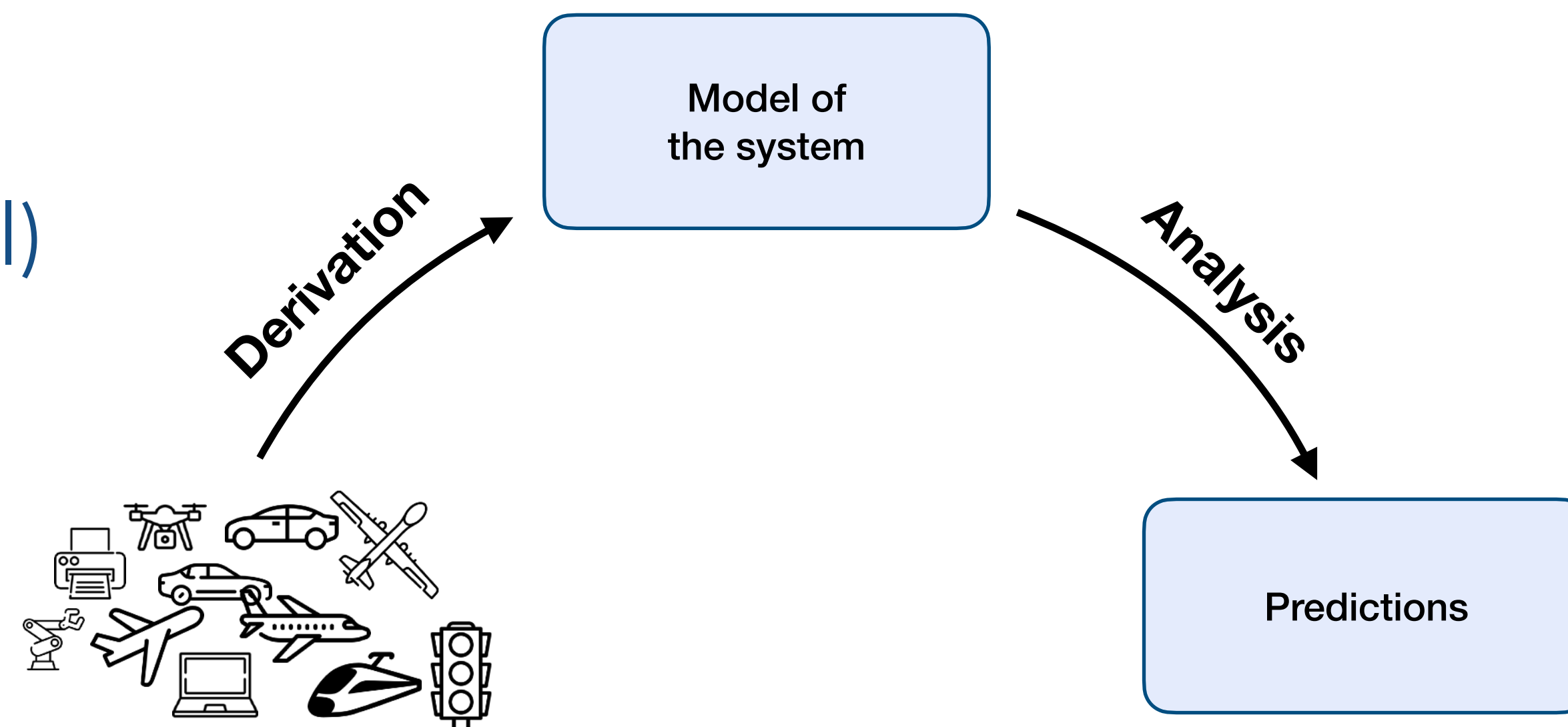


THE BIG PICTURE

Worst-Case Execution
Time (WCET)

To get the predictions right, we need:

- Model with the right **specification**
 - *E.g.*, model must include WCETs to allow (classical) response-time analysis
- Correct model **derivation**
 - Optimistic WCET bound \implies Wrong predictions
- Correct **analysis**
 - Flawed analysis \implies Wrong predictions

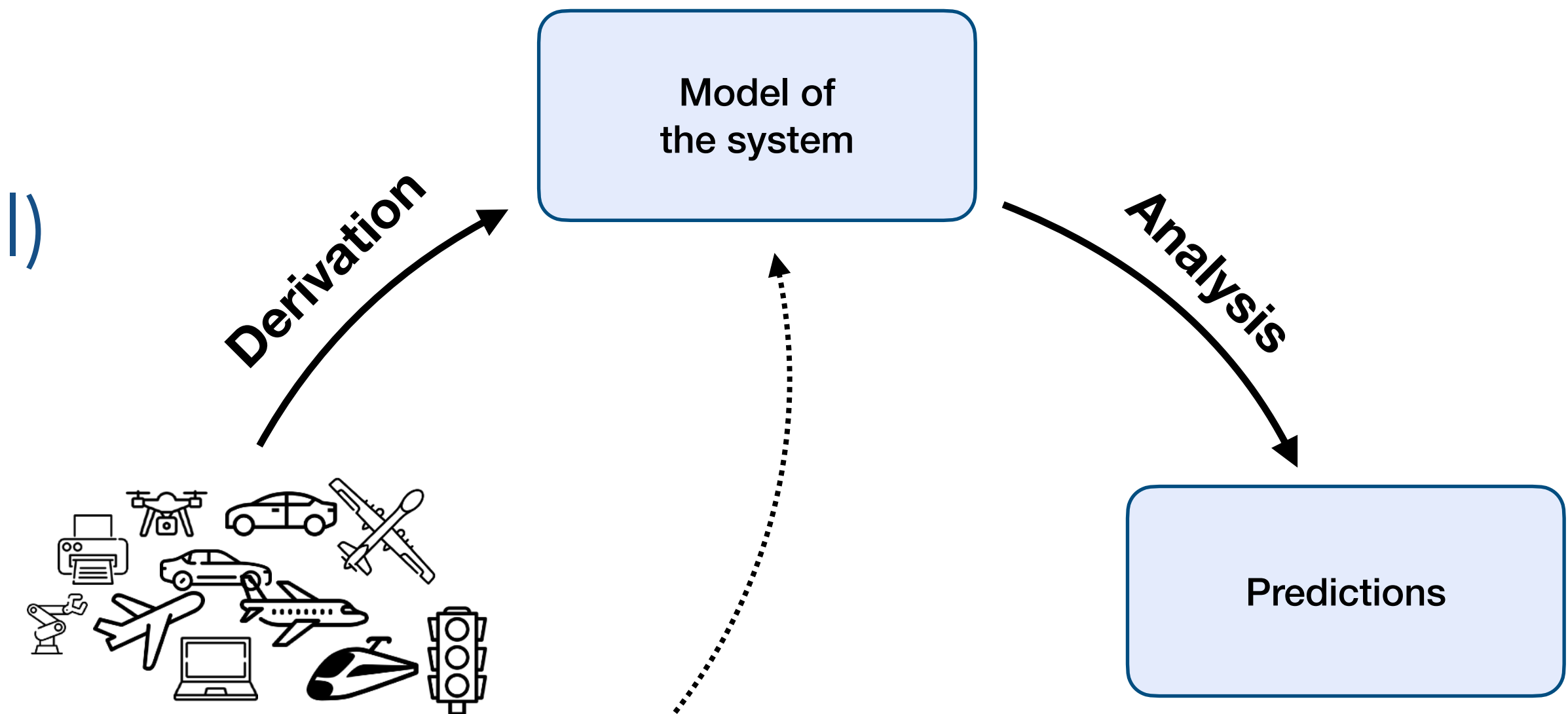


THE BIG PICTURE

Worst-Case Execution
Time (WCET)

To get the predictions right, we need:

- Model with the right **specification**
 - *E.g.*, model must include WCETs to allow (classical) response-time analysis
- Correct model **derivation**
 - Optimistic WCET bound \implies Wrong predictions
- Correct **analysis**
 - Flawed analysis \implies Wrong predictions

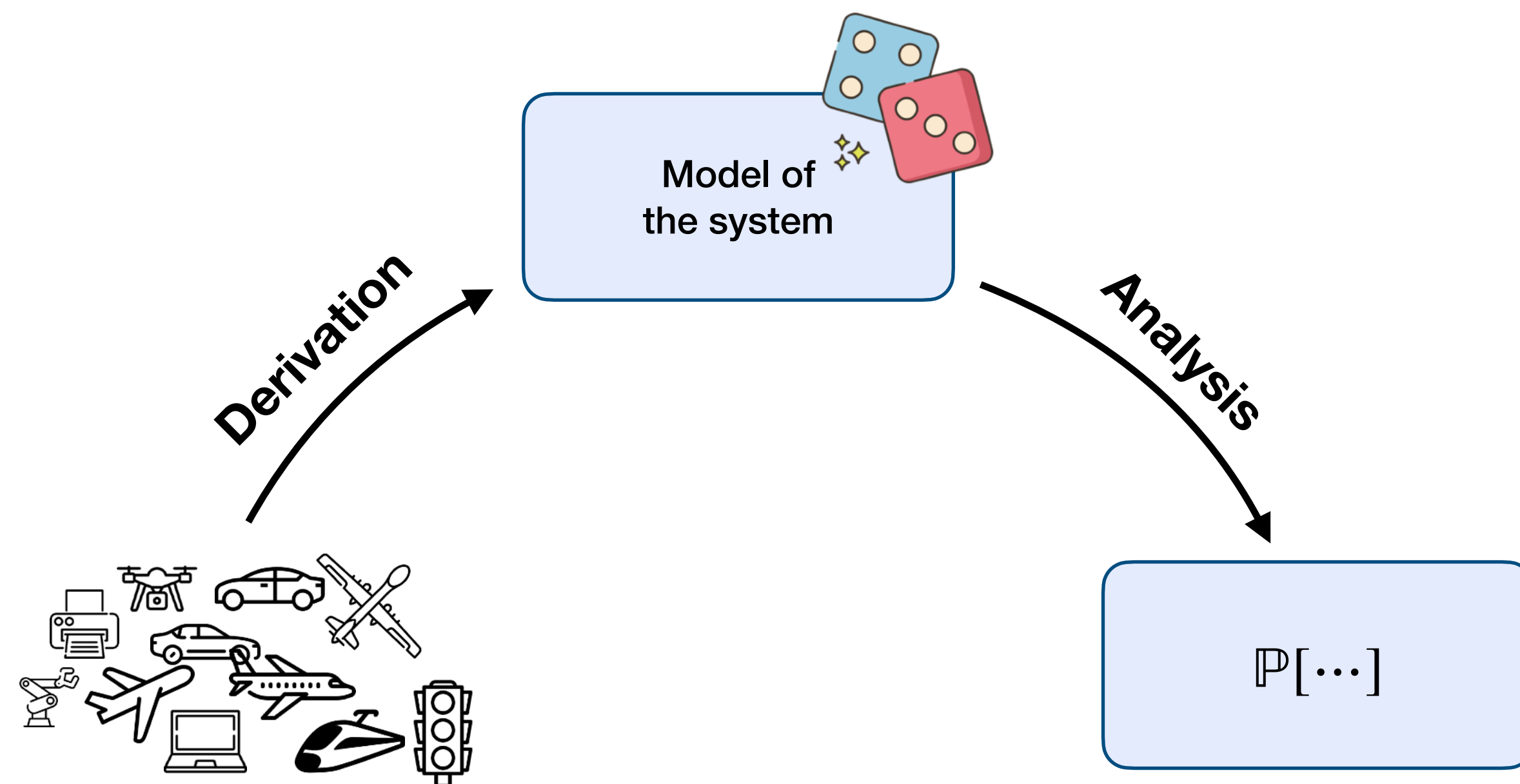


Interpretation of common models is pretty straightforward in the deterministic case

SPECIFICATIONS ARE LESS OBVIOUS IN THE STOCHASTIC CASE

To get the predictions right, we need:

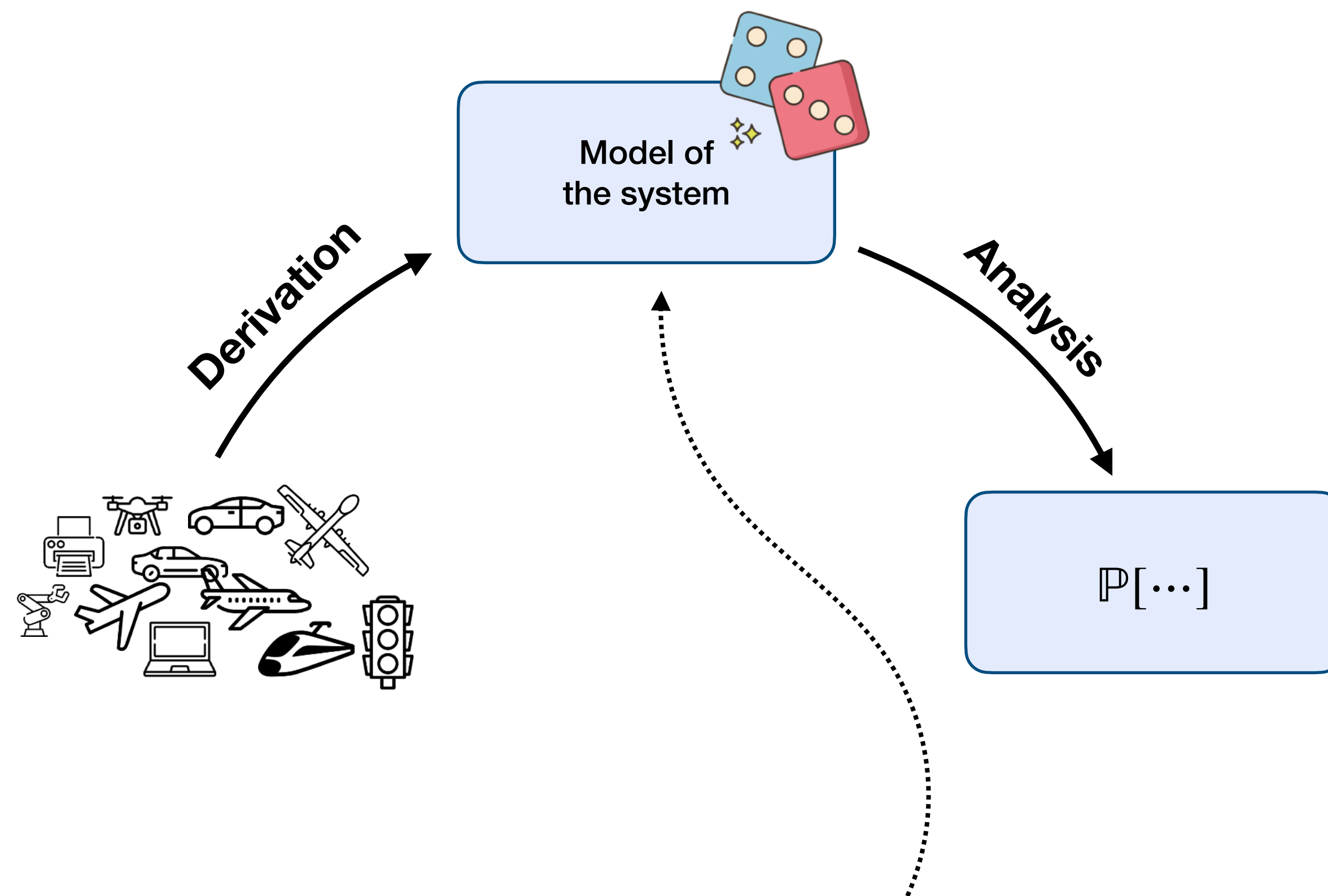
- Model with the right **specification**
 - *E.g.*, model must include ~~WCETs~~ **???** to allow (classical probabilistic) response-time analysis
- Correct model **derivation**
 - Optimistic WCET bound \implies Wrong predictions
- Correct **analysis**
 - Flawed analysis \implies Wrong predictions



SPECIFICATIONS ARE LESS OBVIOUS IN THE STOCHASTIC CASE

To get the predictions right, we need:

- Model with the right **specification**
 - *E.g.*, model must include ~~WCETs~~ **???** to allow (classical probabilistic) response-time analysis
- Correct model **derivation**
 - Optimistic WCET bound \implies Wrong predictions
- Correct **analysis**
 - Flawed analysis \implies Wrong predictions



Specifications of stochastic RTs are much less straightforward

THE CASE FOR STOCHASTIC RTS

Stochastic real-time systems

→ Model of RTSs, where workload parameters are modelled *stochastically*

THE CASE FOR STOCHASTIC RTS

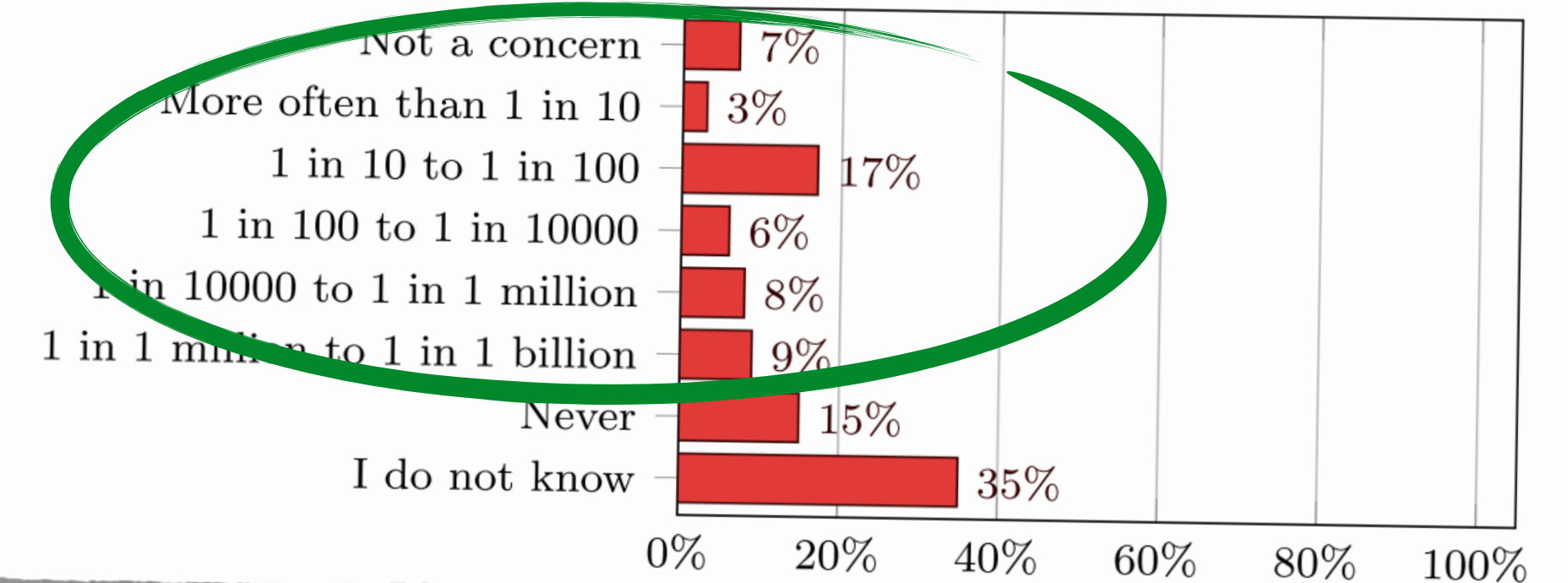
Stochastic real-time systems

→ Model of RTSs, where workload parameters are modelled *stochastically*

Pros:

→ Most systems **can tolerate deadline misses**
 ⇒ Want to take advantage of this

Question 14 For the most time-critical functions in the system, roughly how frequently can the deadline of a function be missed without causing a system failure. (n = 101)



[1]

[1] Akesson, Benny, et al. "A comprehensive survey of industry practice in real-time systems."

THE CASE FOR STOCHASTIC RTS

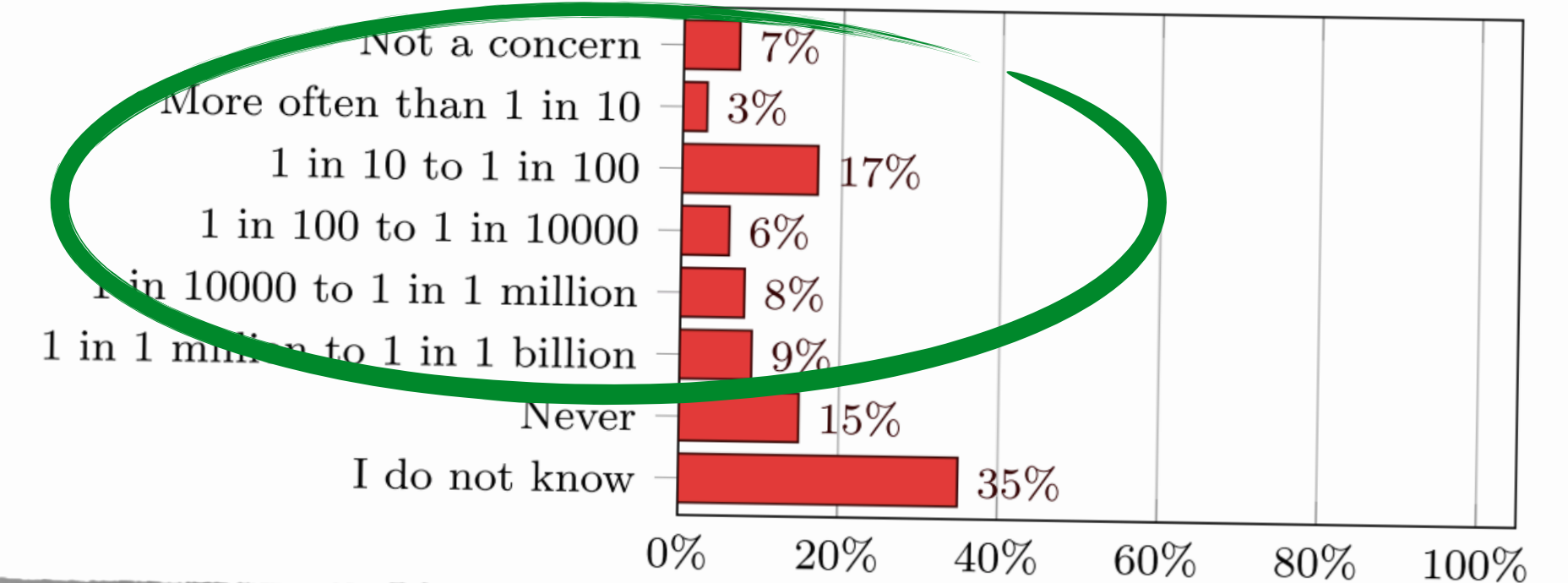
Stochastic real-time systems

→ Model of RTSs, where workload parameters are modelled *stochastically*

Pros:

- Most systems **can tolerate deadline misses**
 - ⇒ Want to take advantage of this
- Allows answering **quantitative** questions

Question 14 For the most time-critical functions in the system, roughly how frequently can the deadline of a function be missed without causing a system failure. (n = 101)



[1]

[1] Akesson, Benny, et al. "A comprehensive survey of industry practice in real-time systems."

[2] Rivas, Juan M., et al. "Calculating latencies in an engine management system using response time analysis with MAST."

THE CASE FOR STOCHASTIC RTS

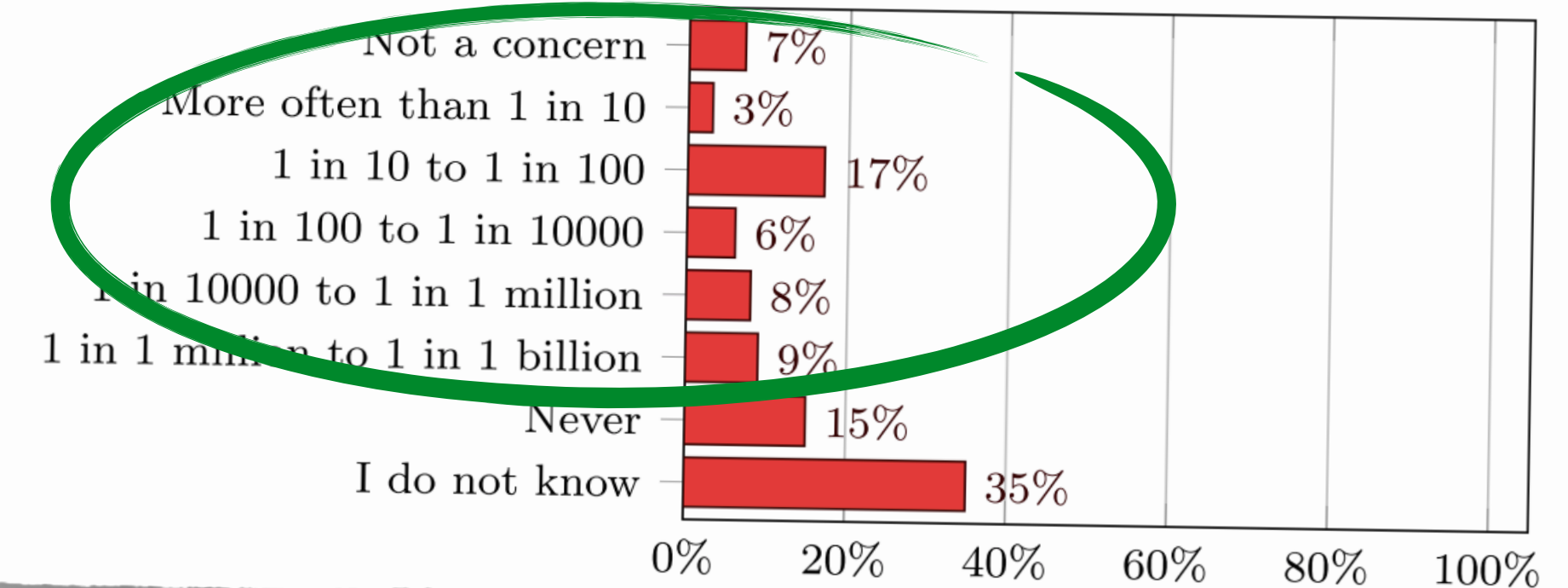
Stochastic real-time systems

→ Model of RTSs, where workload parameters are modelled *stochastically*

Pros:

- Most systems **can tolerate deadline misses**
 - ⇒ Want to take advantage of this
- Allows answering **quantitative** questions
- Enables analysis of transiently **overloaded systems**
 - Ubiquitous in practice
 - *E.g., FMTV Challenge 2016*

Question 14 For the most time-critical functions in the system, roughly how frequently can the deadline of a function be missed without causing a system failure. (n = 101)



[1]

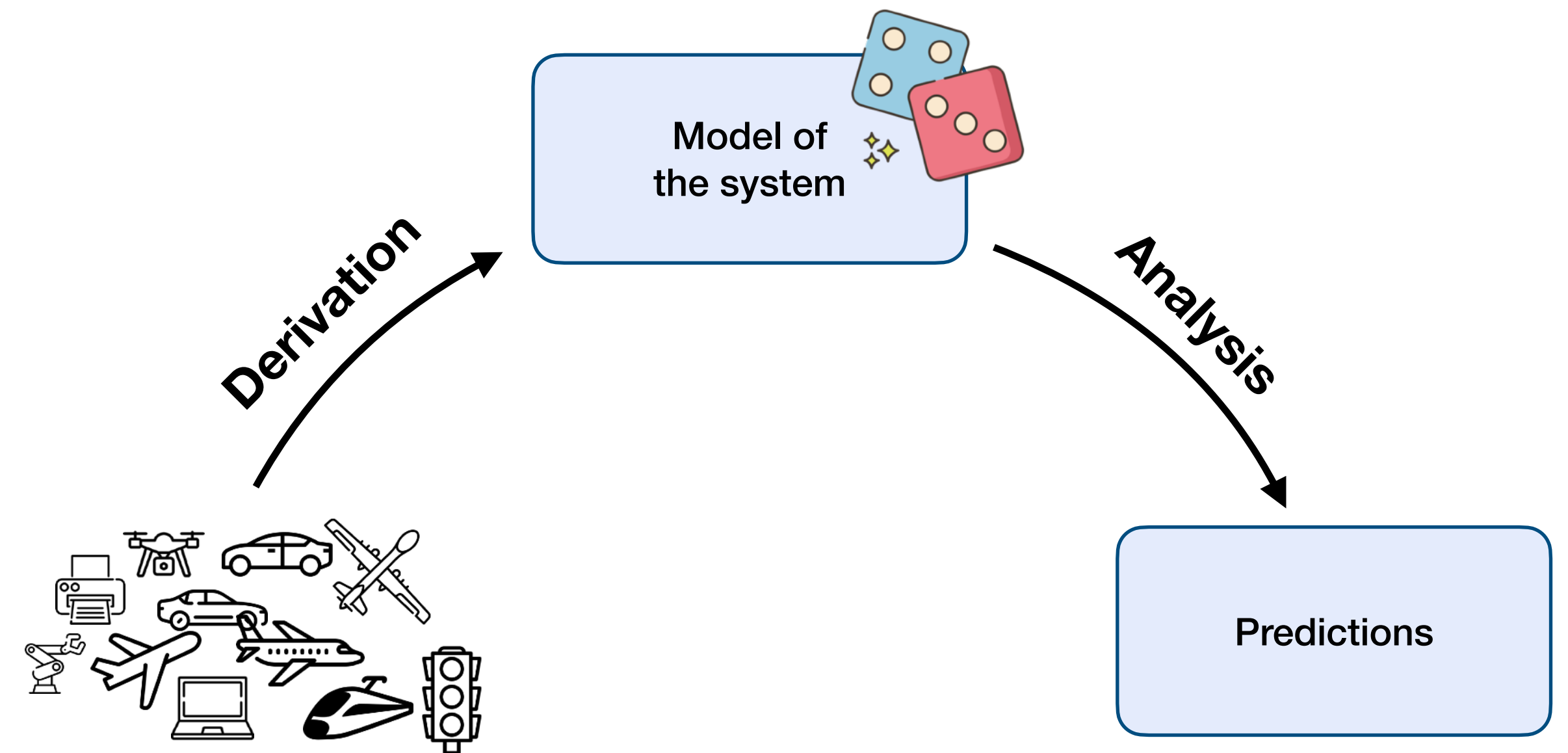
The total utilization of that system goes above 100%. Using response time analysis in such situation automatically yields unbounded (infinite) worst-case response times. [2]

[1] Akesson, Benny, et al. "A comprehensive survey of industry practice in real-time systems."

[2] Rivas, Juan M., et al. "Calculating latencies in an engine management system using response time analysis with MAST."

DEPENDENCY IN STOCHASTIC RTS

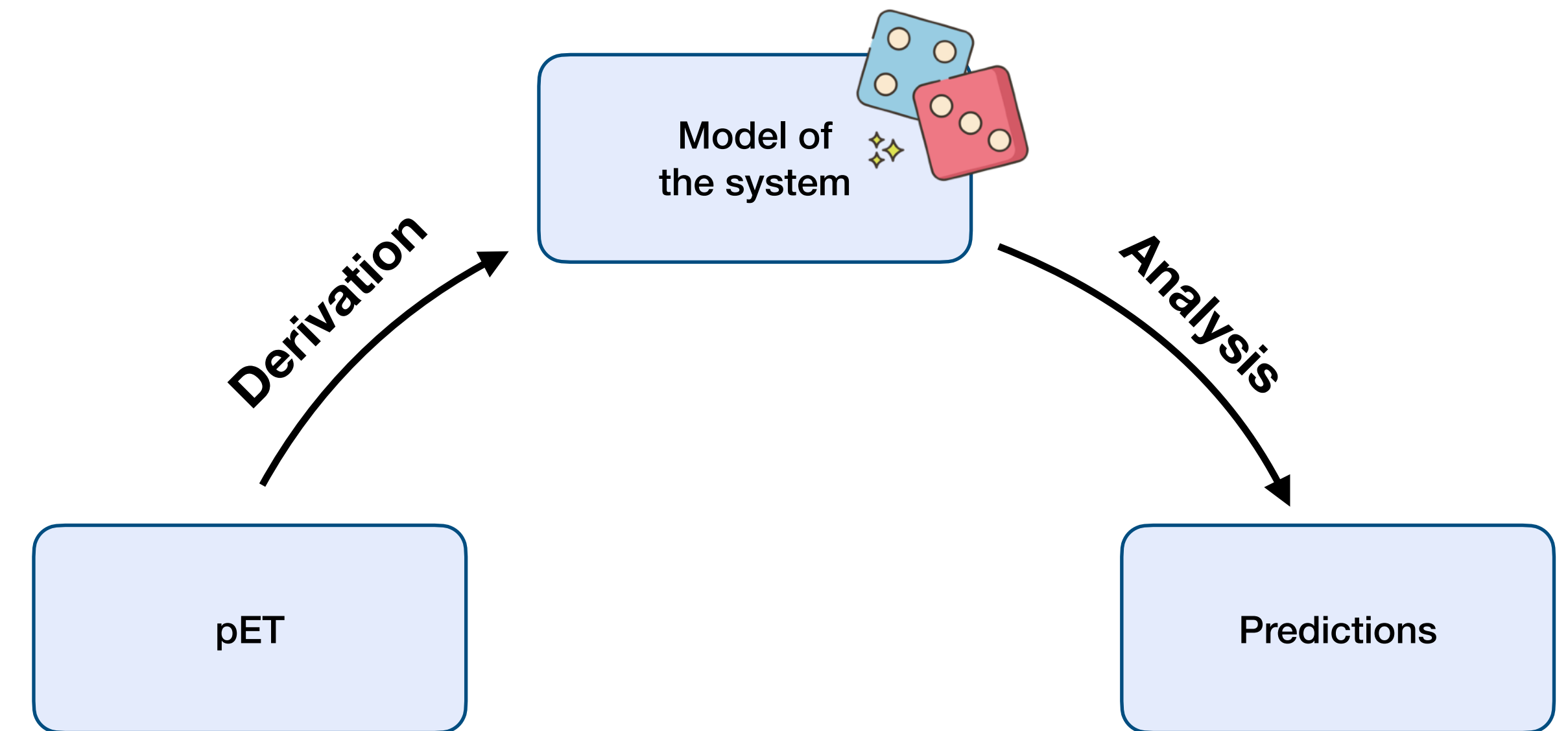
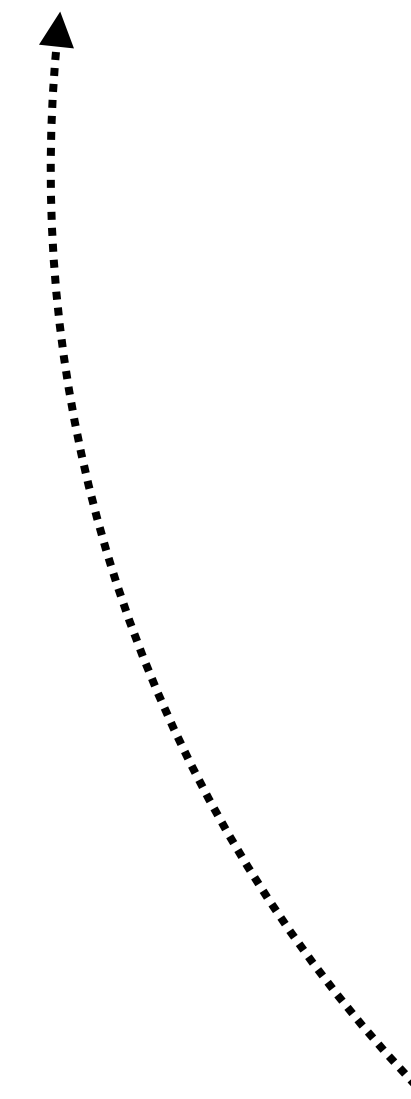
THE PROBLEM OF STOCHASTIC RTS: DEPENDENCY



THE PROBLEM OF STOCHASTIC RTS: DEPENDENCY

Probabilistic Execution Times (pETs)

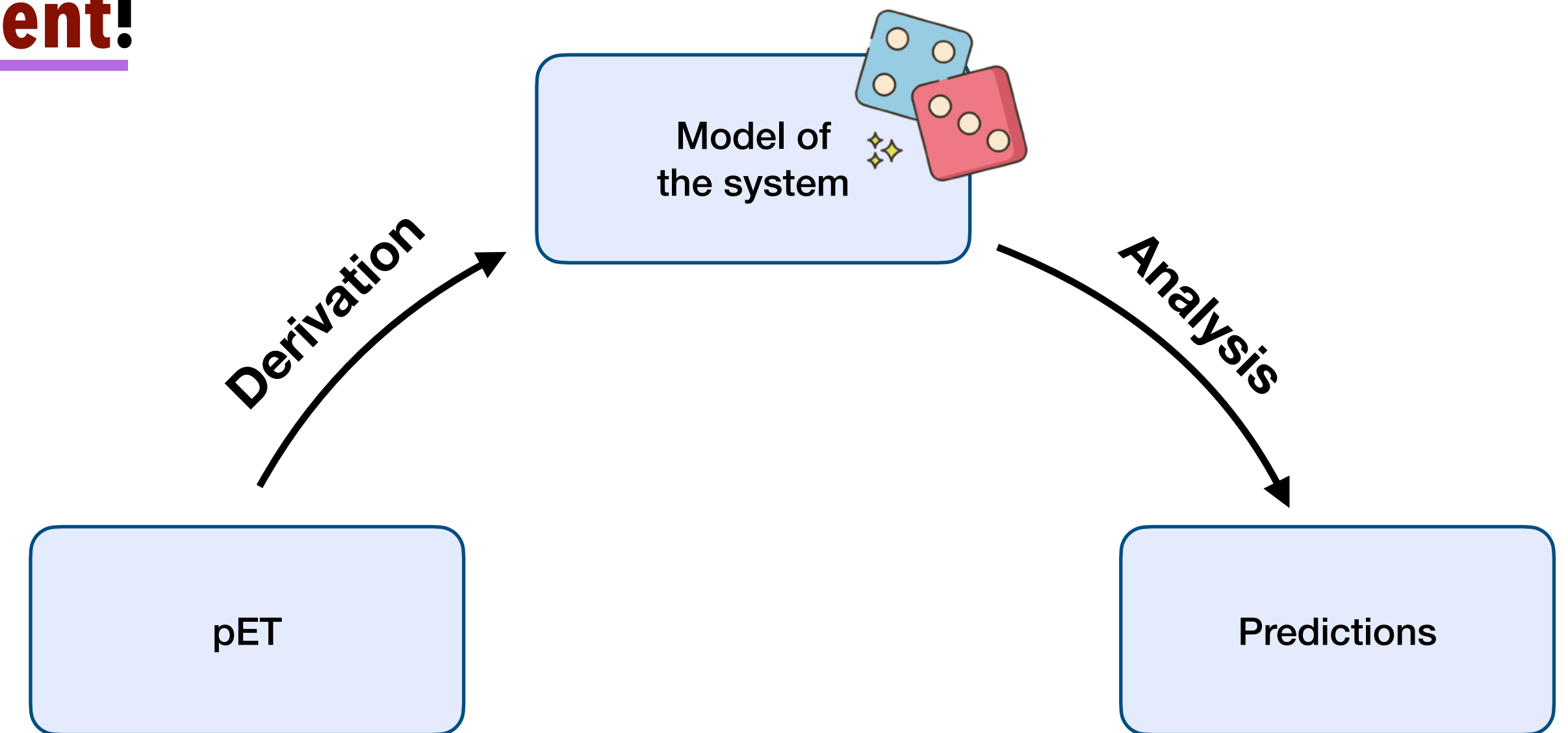
Ground-truth behavior
of jobs in the system



THE PROBLEM OF STOCHASTIC RTS: DEPENDENCY

Probabilistic Execution Times (pETs) are **dependent!**

Ground-truth behavior
of jobs in the system

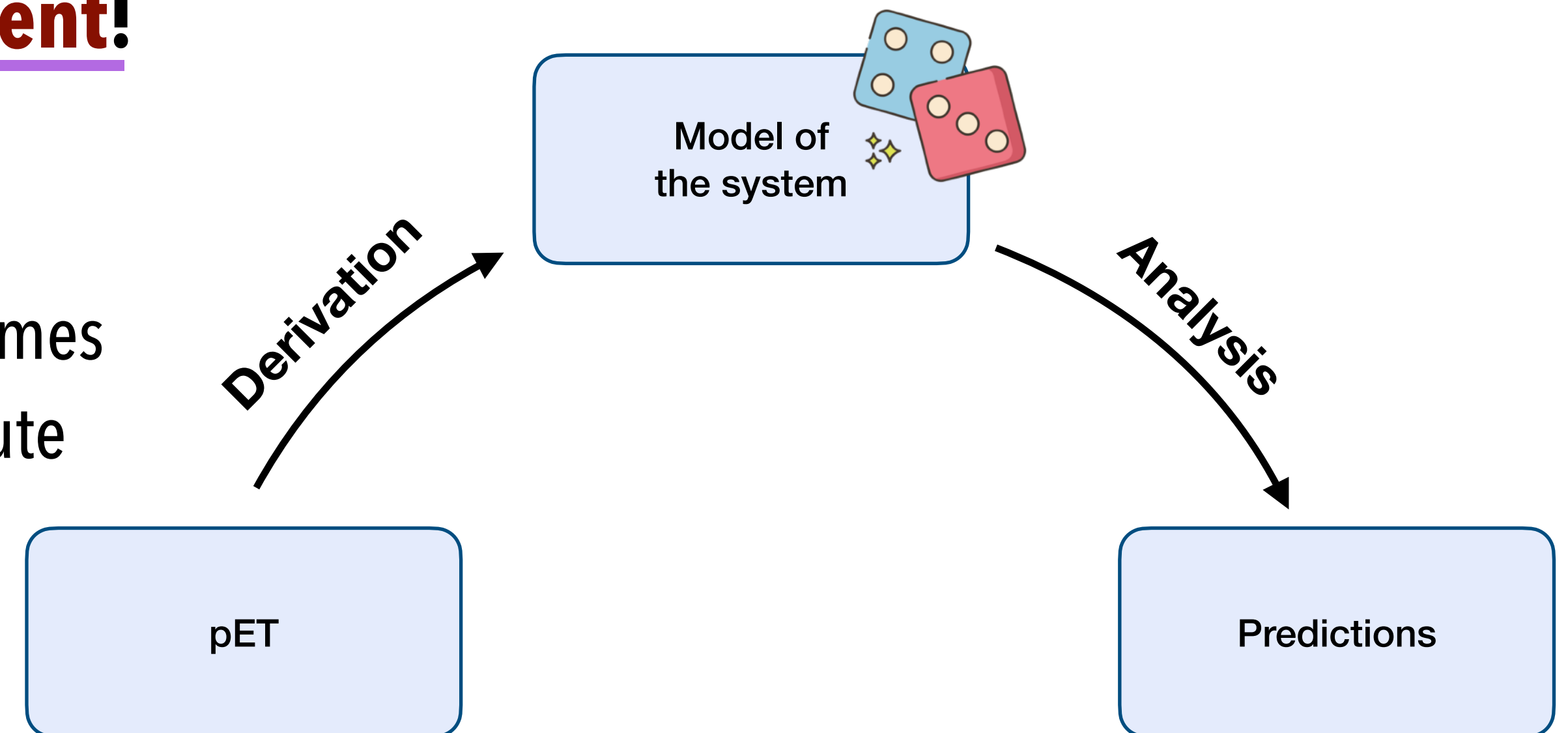


THE PROBLEM OF STOCHASTIC RTS: DEPENDENCY

Probabilistic Execution Times (pETs) are **dependent!**

- ▶ Image processing: Two consecutive frames might take similar amounts of compute

Ground-truth behavior
of jobs in the system

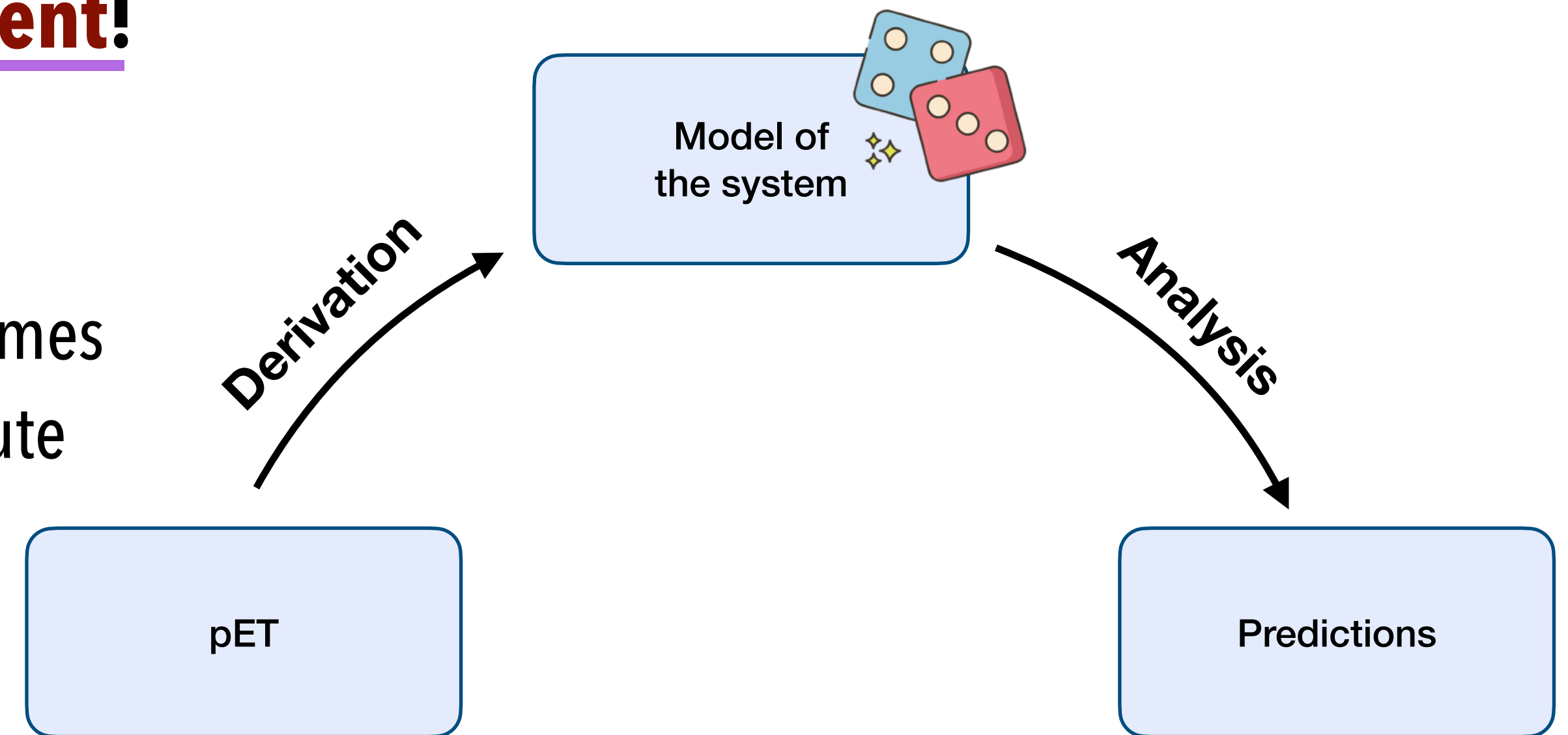


THE PROBLEM OF STOCHASTIC RTS: DEPENDENCY

Probabilistic Execution Times (pETs) are **dependent!**

- ▶ Image processing: Two consecutive frames might take similar amounts of compute
- ▶ Behavior of a prior job influences the state of the cache

Ground-truth behavior of jobs in the system

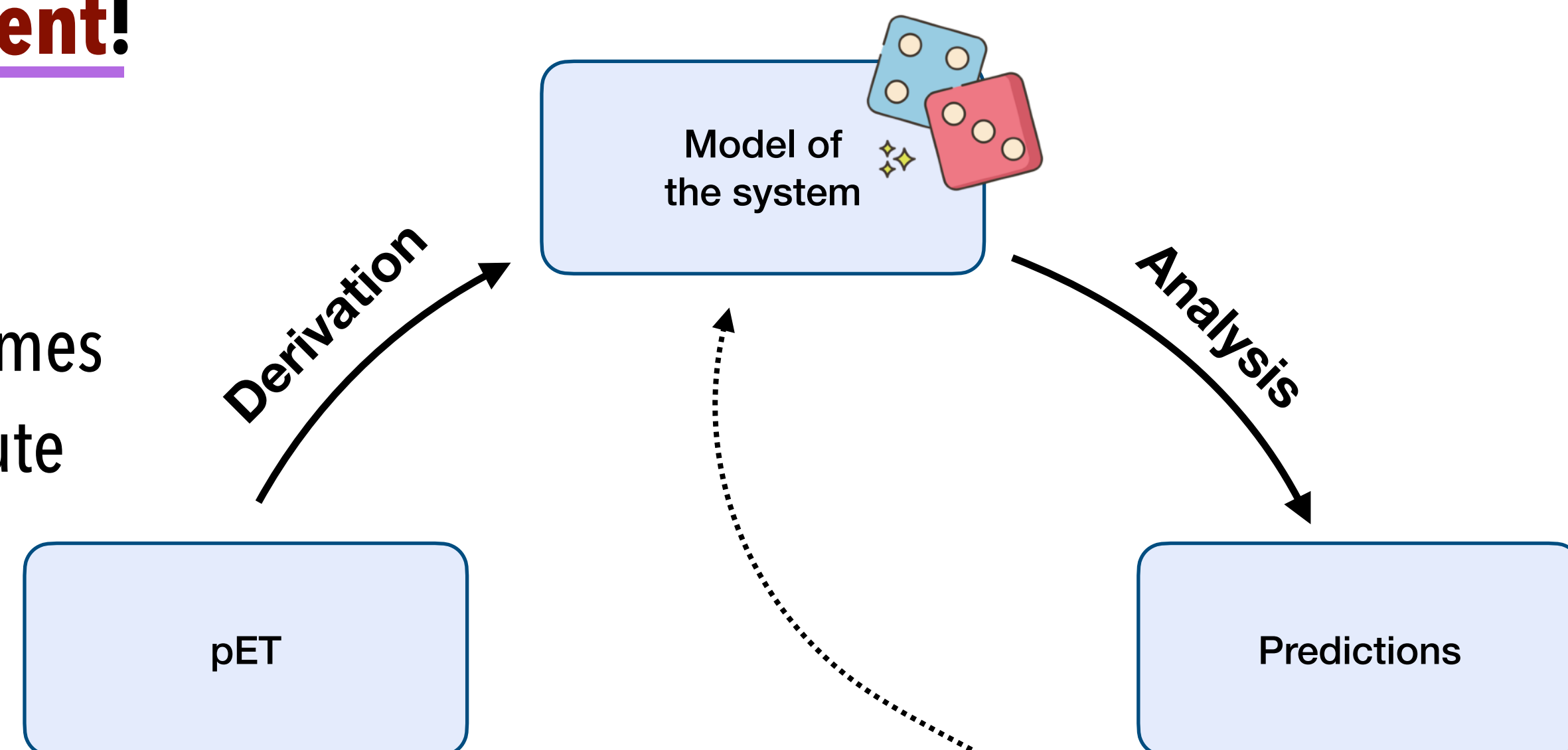


THE PROBLEM OF STOCHASTIC RTS: DEPENDENCY

Probabilistic Execution Times (pETs) are **dependent!**

- ▶ Image processing: Two consecutive frames might take similar amounts of compute
- ▶ Behavior of a prior job influences the state of the cache

Ground-truth behavior of jobs in the system



Q: Can we disregard dependency and continue anyway?

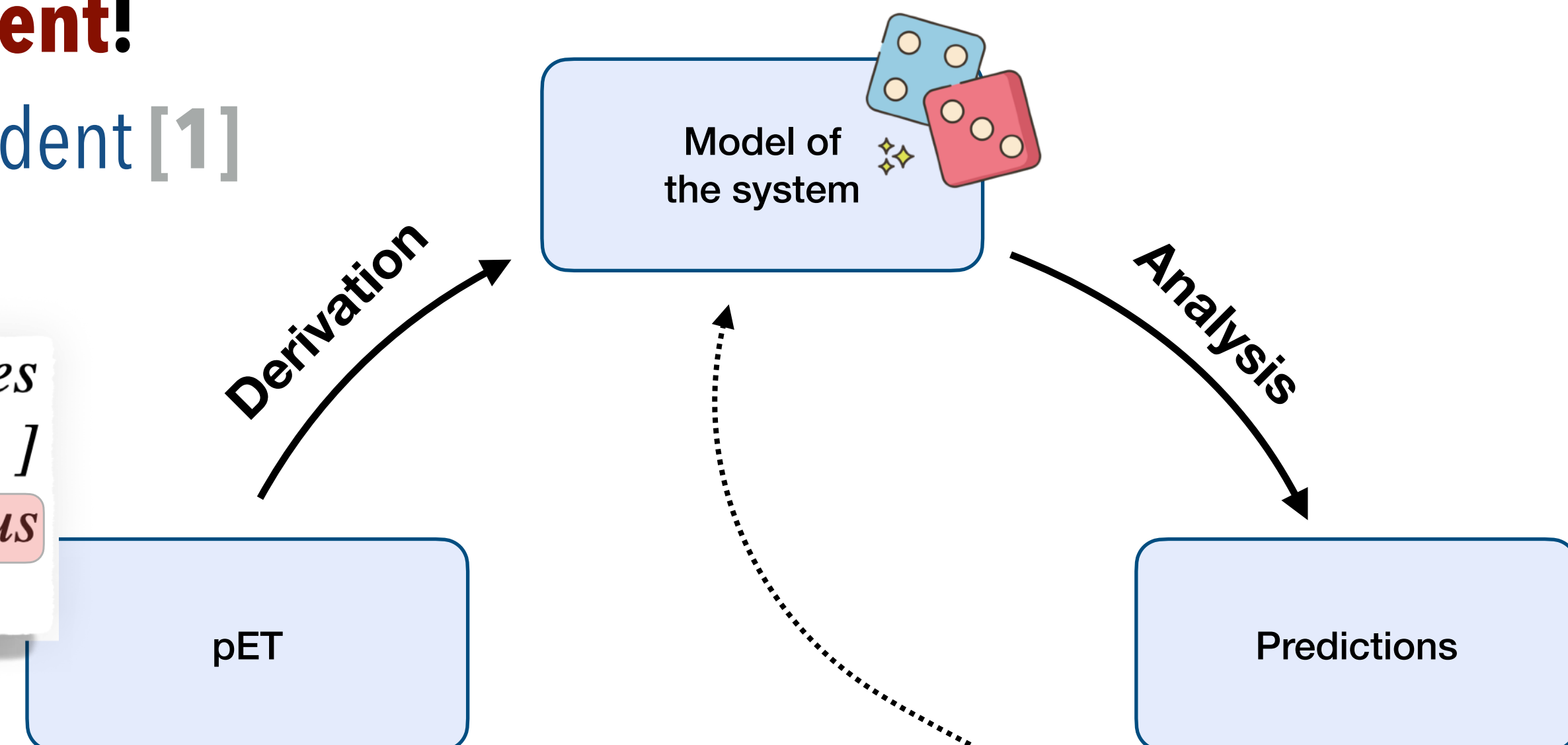
THE PROBLEM OF STOCHASTIC RTS: DEPENDENCY

Probabilistic Execution Times (pETs) are **dependent!**

→ Tia *et al.* 1995: computation times are **not** independent [1]

→ Ignoring this fact may lead to incorrect bounds

“Unfortunately, the computation times of individual requests are not statistically independent. [...] As a consequence, the probability of meeting deadlines thus computed may be overly optimistic.”



Q: Can we disregard dependency and continue anyway?

A: No, the results can be optimistic

[1] Tia, T-S., et al. "Probabilistic performance guarantee for real-time tasks with varying computation times."

THE PROBLEM OF STOCHASTIC RTS: DEPENDENCY

Probabilistic Execution Times (pETs) are **dependent!**

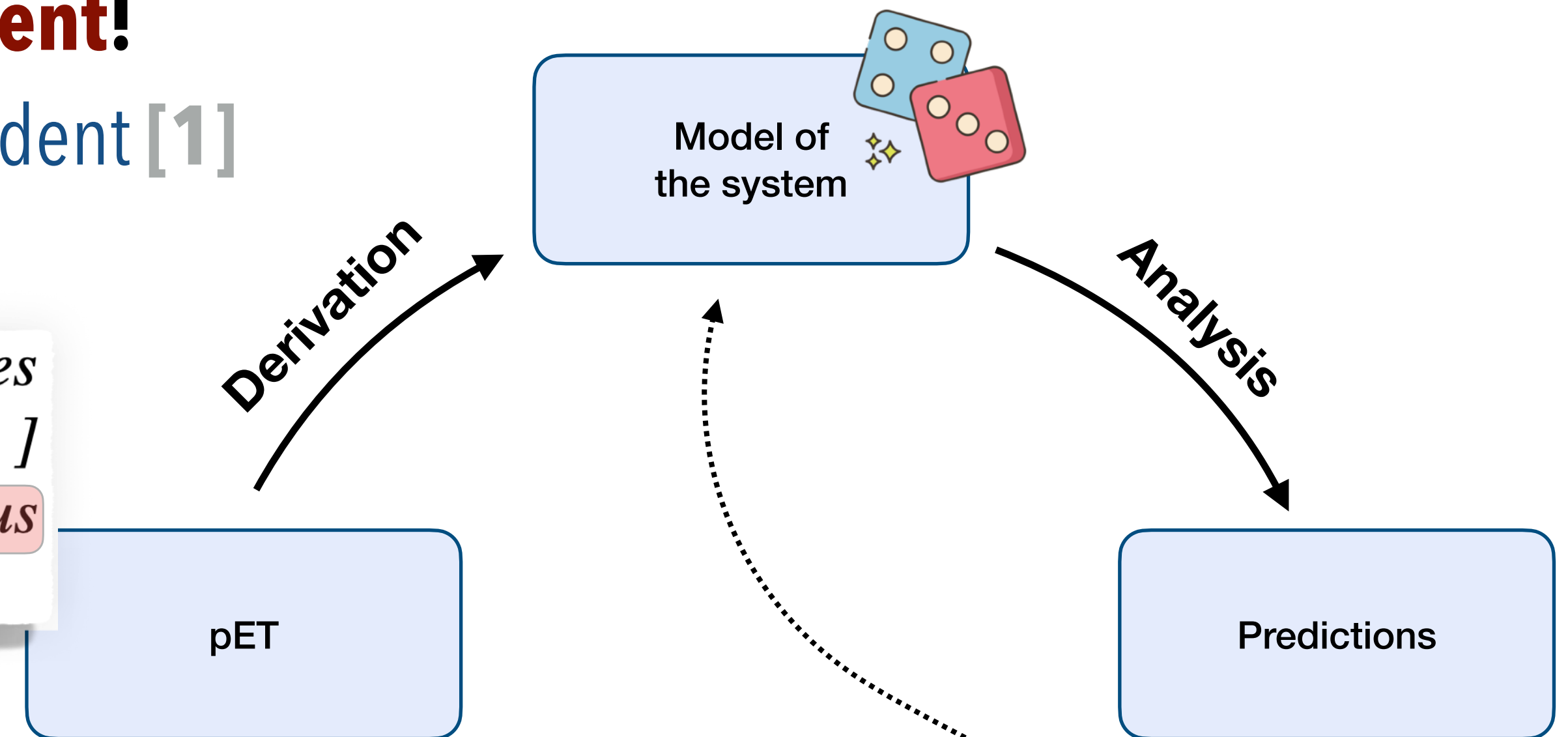
→ Tia *et al.* 1995: computation times are **not** independent [1]

→ Ignoring this fact may lead to incorrect bounds

“Unfortunately, the computation times of individual requests are not statistically independent. [...] As a consequence, the probability of meeting deadlines thus computed may be overly optimistic.”

→ Limits the application of probability theory tools

→ *E.g.*, convolution is not applicable



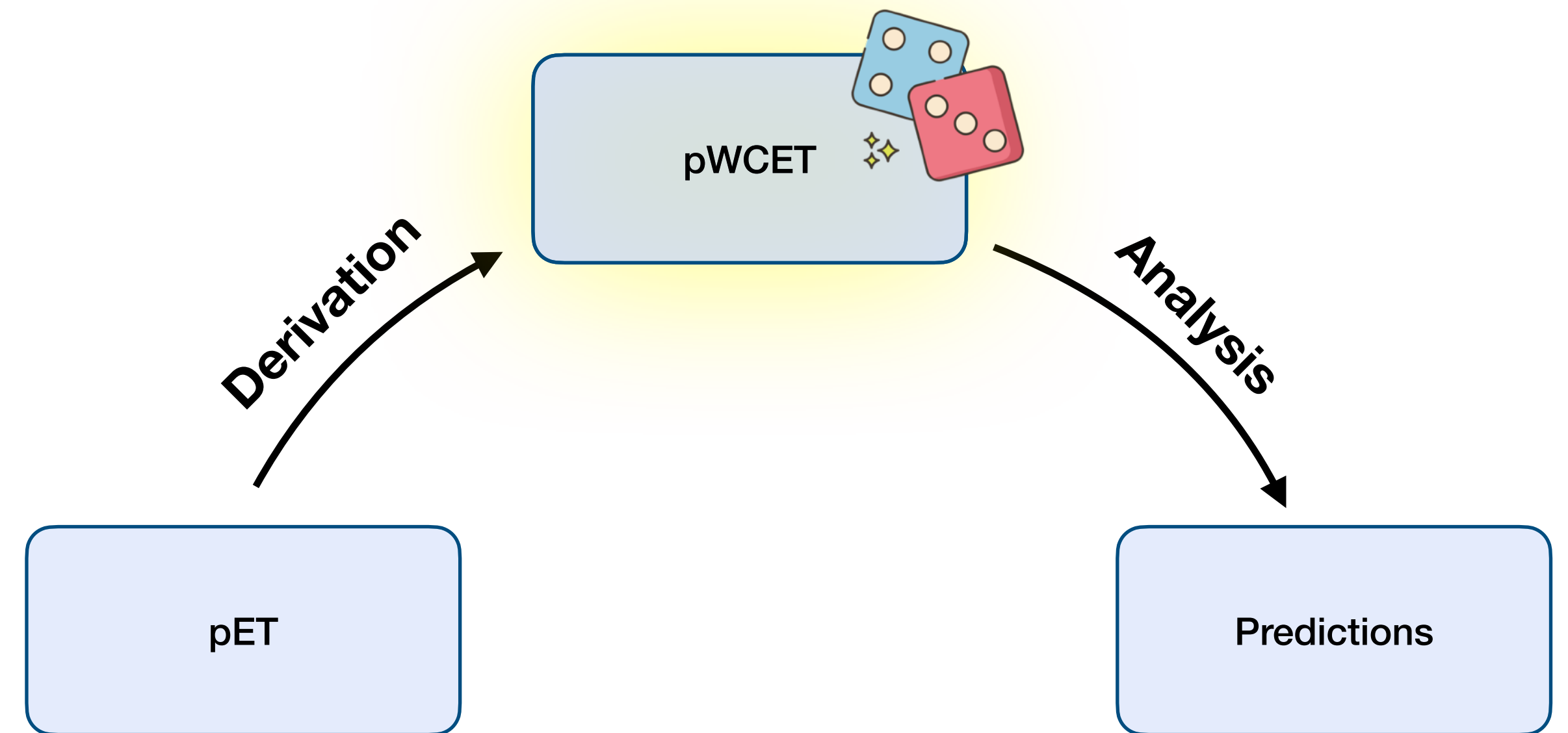
Q: Can we disregard dependency and continue anyway?

A: No, the results can be optimistic

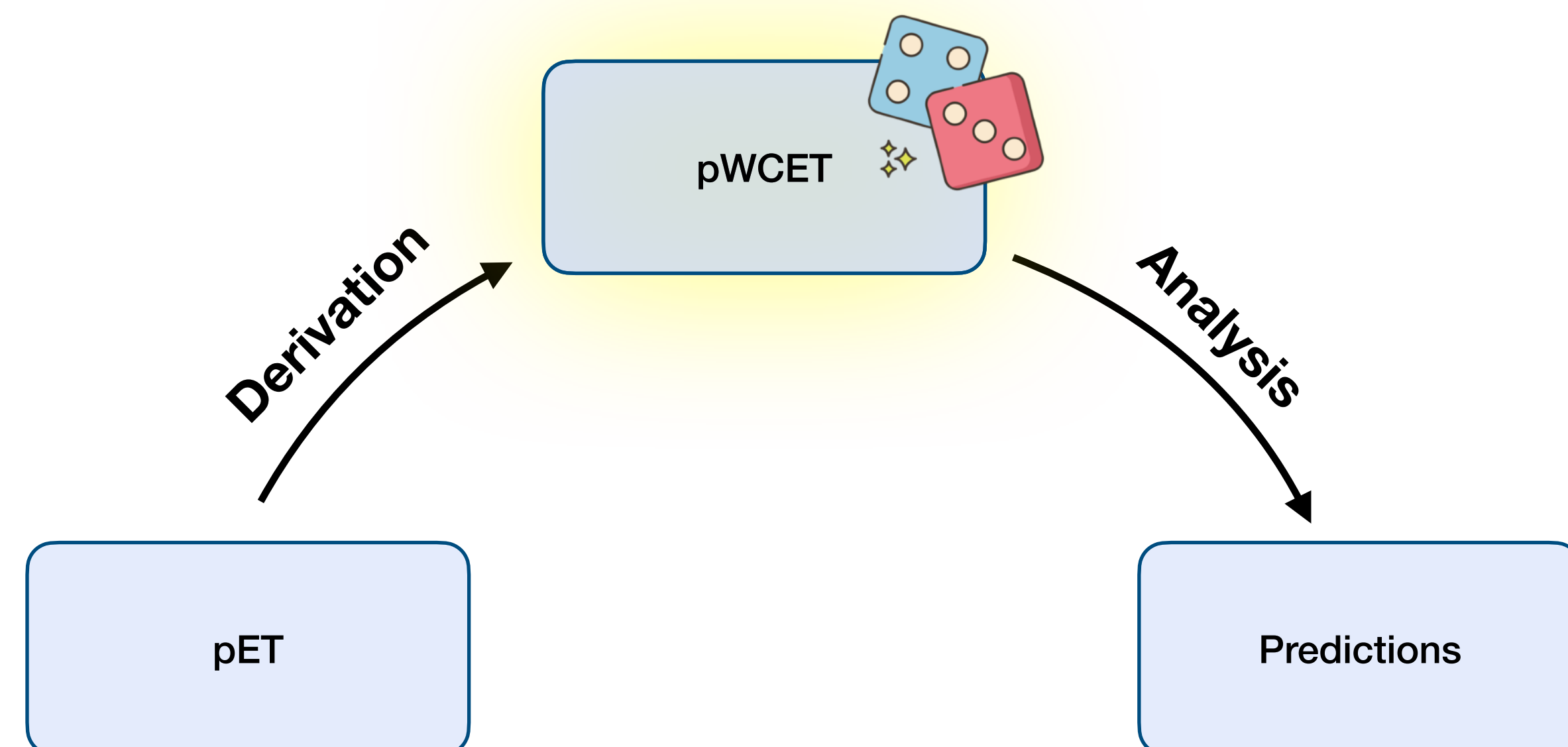
[1] Tia, T-S., et al. "Probabilistic performance guarantee for real-time tasks with varying computation times."

PRIOR WORK: pWCET

PROBABILISTIC WORST-CASE EXECUTION TIME (pWCET)



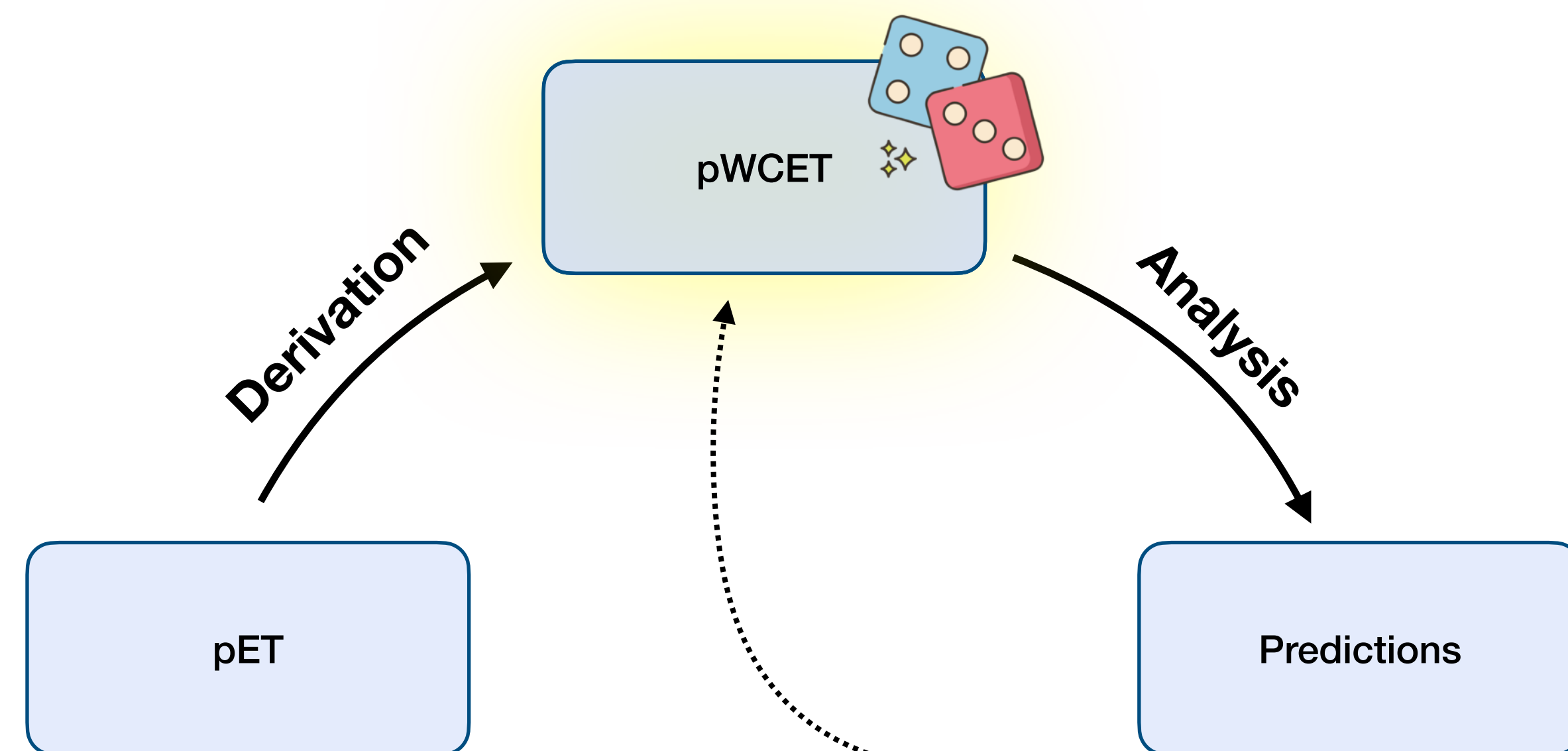
PROBABILISTIC WORST-CASE EXECUTION TIME (pWCET)



We note that the actual execution times for a sequence of jobs of a task, which exercise the same or different paths, may well show **strong correlations and dependences**. It is **the modelling of the execution times via an appropriate pWCET distribution which enables probabilistic independence to be assumed**. (This is similar to the conventional case of a single WCET which can similarly be used in this way, even though the actual execution times of different jobs have strong dependences).

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

PROBABILISTIC WORST-CASE EXECUTION TIME (pWCET)



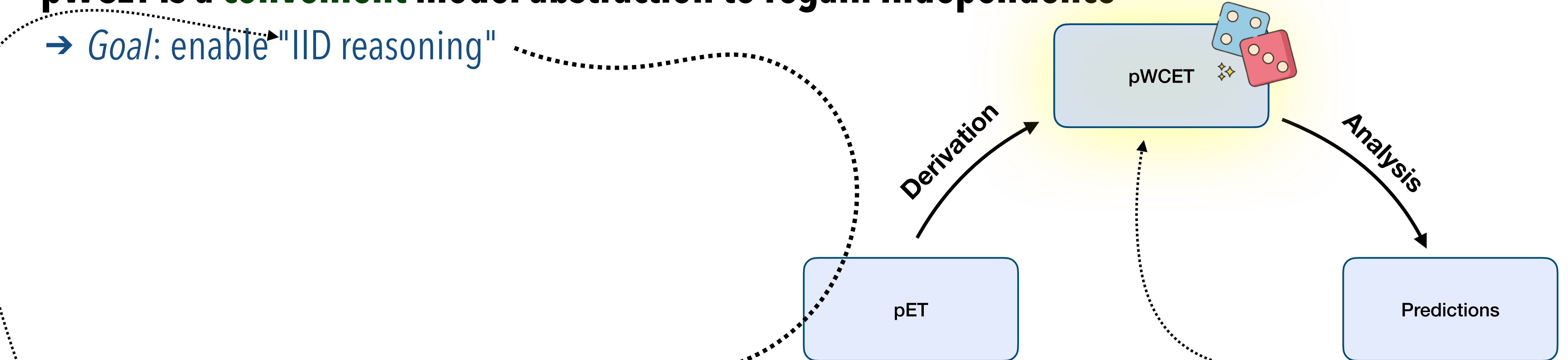
We note that the actual execution times for a sequence of jobs of a task, which exercise the same or different paths, may well show **strong correlations and dependences**. It is **the modelling of the execution times via an appropriate pWCET distribution which enables probabilistic independence to be assumed**. (This is similar to the conventional case of a single WCET which can similarly be used in this way, even though the actual execution times of different jobs have strong dependences).

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

PROBABILISTIC WORST-CASE EXECUTION TIME (pWCET)

pWCET is a **convenient** model abstraction to regain independence

→ Goal: enable "IID reasoning"



Independent and identically distributed

We note that the actual execution times for a sequence of jobs of a task, which exercise the same or different paths, may well show **strong correlations and dependences**. It is **the modelling of the execution times via an appropriate pWCET distribution which enables probabilistic independence to be assumed**. (This is similar to the conventional case of a single WCET which can similarly be used in this way, even though the actual execution times of different jobs have strong dependences).

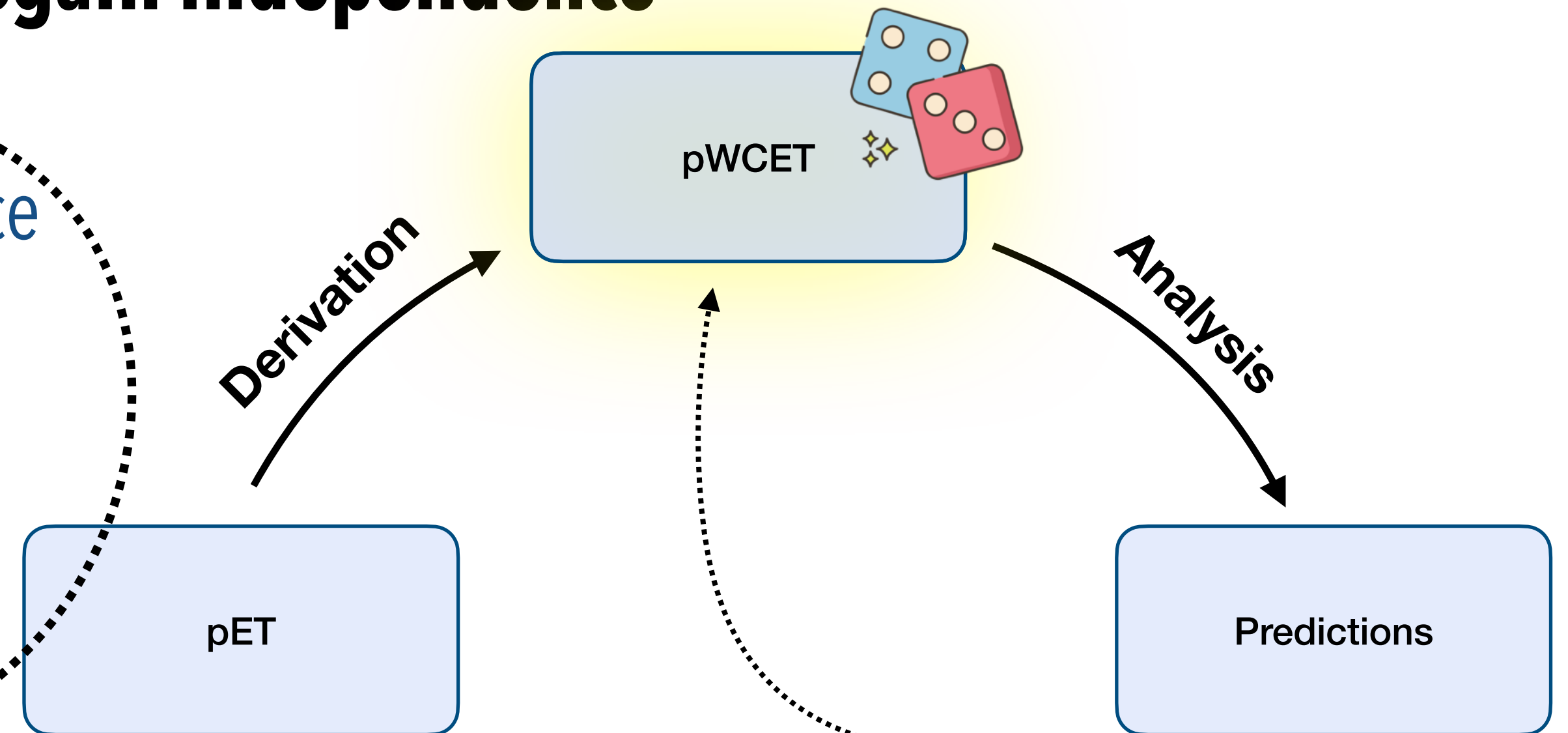
[1]

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

PROBABILISTIC WORST-CASE EXECUTION TIME (pWCET)

pWCET is a convenient model abstraction to regain independence

- Goal: enable "IID reasoning"
- The **mainstream** approach to hiding dependence
- Unlocks powerful probability theory techniques
 - Such as *convolution, Chernoff bound, etc.*



Independent and identically distributed

We note that the actual execution times for a sequence of jobs of a task, which exercise the same or different paths, may well show **strong correlations and dependences**. It is **the modelling of the execution times via an appropriate pWCET distribution which enables probabilistic independence to be assumed**. (This is similar to the conventional case of a single WCET which can similarly be used in this way, even though the actual execution times of different jobs have strong dependences).

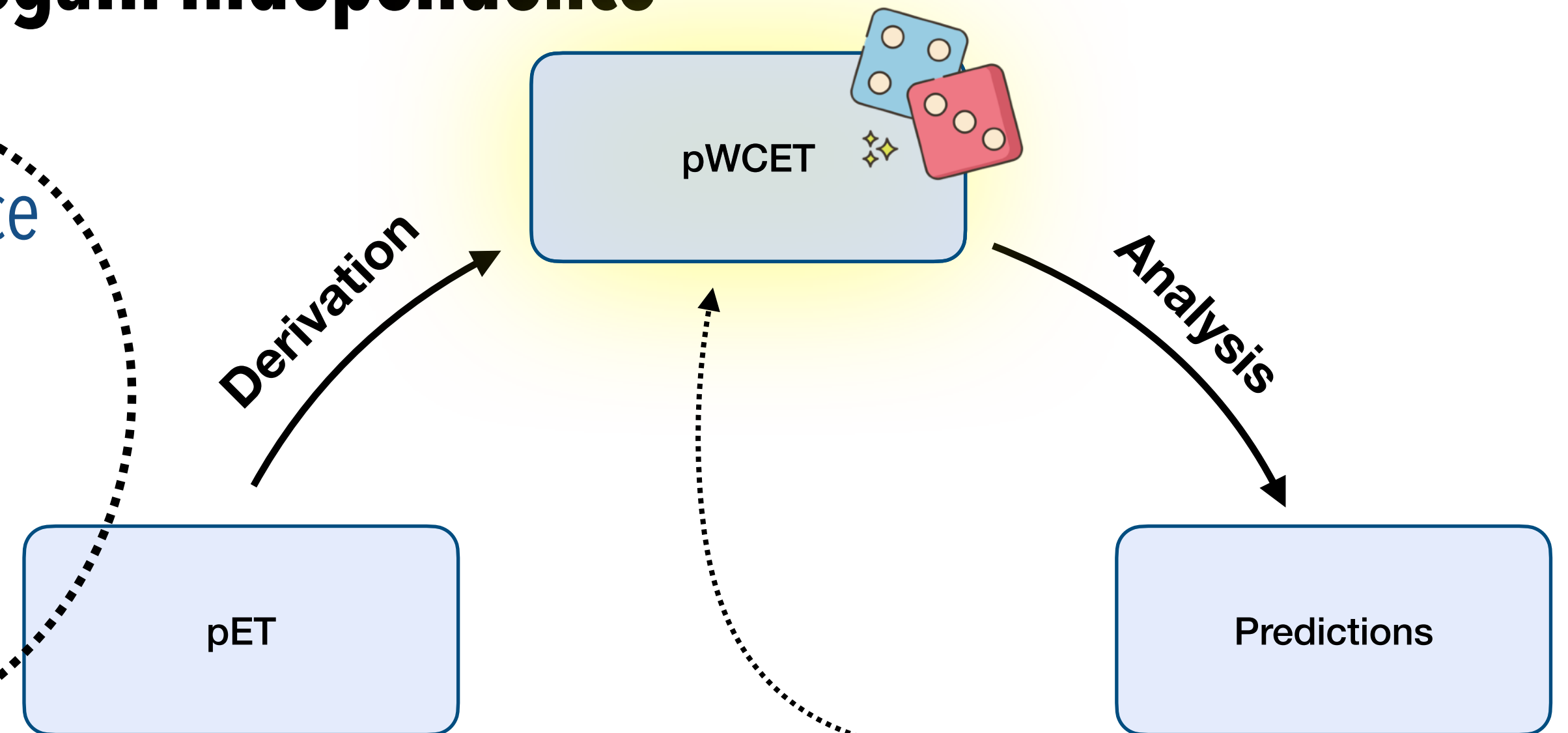
[1]

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

PROBABILISTIC WORST-CASE EXECUTION TIME (pWCET)

pWCET is a convenient model abstraction to regain independence

- Goal: enable "IID reasoning"
- The **mainstream** approach to hiding dependence
- Unlocks powerful probability theory techniques
 - Such as *convolution, Chernoff bound, etc.*
- but when **exactly** is a pWCET distribution "appropriate"?



Independent and identically distributed

We note that the actual execution times for a sequence of jobs of a task, which exercise the same or different paths, may well show strong correlations and dependences. It is the modelling of the execution times via an appropriate pWCET distribution which enables probabilistic independence to be assumed. (This is similar to the conventional case of a single WCET which can similarly be used in this way, even though the actual execution times of different jobs have strong dependences).

[1]

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

THE STATE-OF-THE-ART pWCET DEFINITION

► **Definition 2.** The *probabilistic Worst-Case Execution Time (pWCET) distribution* for a task is the least upper bound, in the sense of the greater than or equal to operator \succeq (defined below), on the execution time distribution of the jobs of the task for every valid *scenario of operation*, where a *scenario of operation* is defined as an infinitely repeating sequence of input states (including both input values and software state variables) and initial hardware states that characterise a feasible way in which recurrent execution of the task may occur. [1]

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

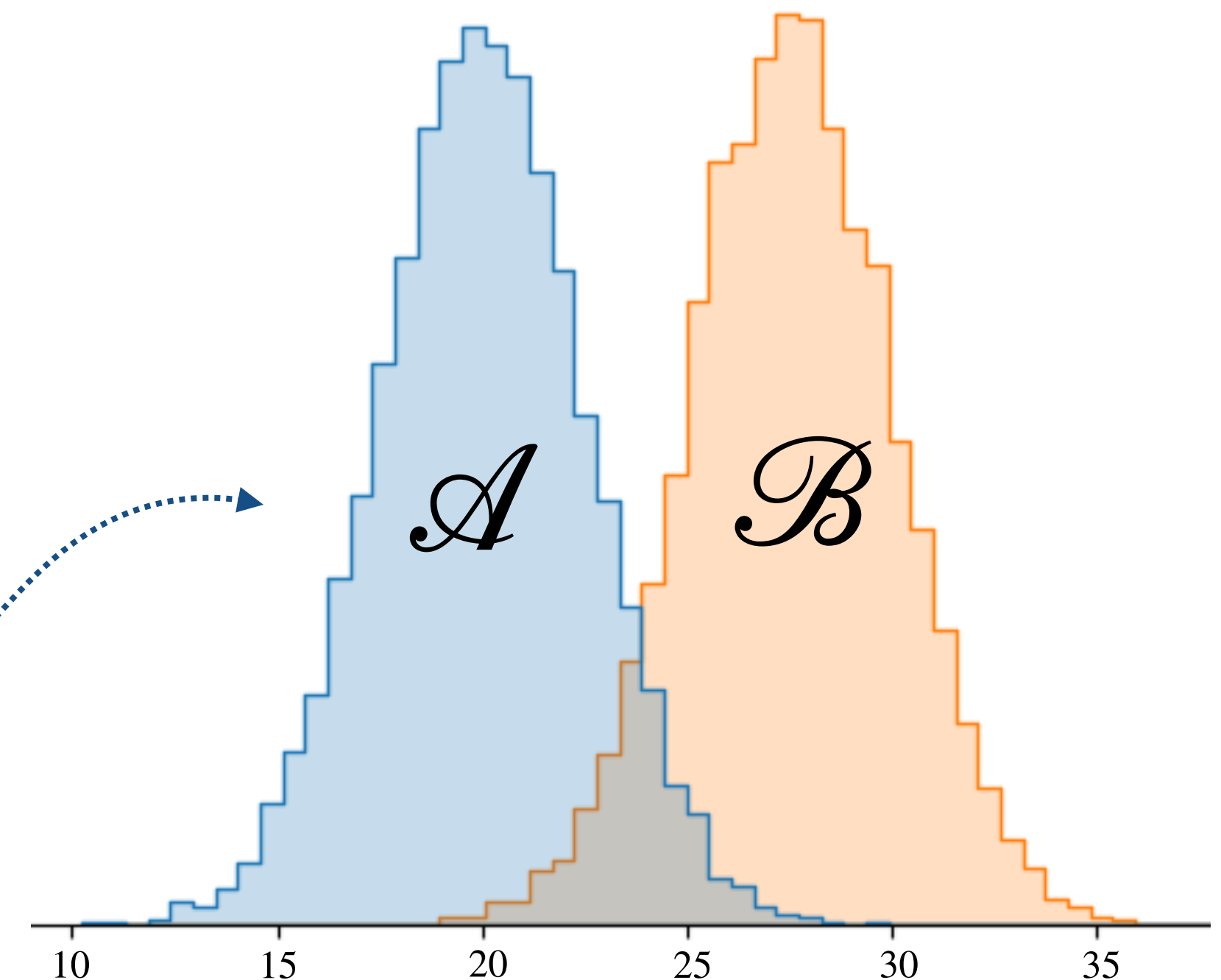
THE STATE-OF-THE-ART pWCET DEFINITION

► **Definition 2.** The *probabilistic Worst-Case Execution Time (pWCET) distribution* for a task is the least upper bound, in the sense of the greater than or equal to operator \succeq (defined below), on the execution time distribution of the jobs of the task for every valid *scenario of operation*, where a *scenario of operation* is defined as an infinitely repeating sequence of input states (including both input values and software state variables) and initial hardware states that characterise a feasible way in which recurrent execution of the task may occur. [1]



Side note: dominance relation \preceq

- Proposed by Diaz *et al.* in 2004 [2]
- Partial order on random variables
- Similar to stochastic dominance
- $A \preceq B := \forall x, \mathbb{P}[A \leq x] \geq \mathbb{P}[B \leq x]$



[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

[2] Diaz, Jose Luis, et al. "Pessimism in the stochastic analysis of real-time systems: Concept and applications."

THE STATE-OF-THE-ART pWCET DEFINITION

► **Definition 2.** The *probabilistic Worst-Case Execution Time (pWCET) distribution* for a task is the least upper bound, in the sense of the greater than or equal to operator \succeq (defined below), on the execution time distribution of the jobs of the task for every valid *scenario of operation*, where a *scenario of operation* is defined as an infinitely repeating sequence of input states (including both input values and software state variables) and initial hardware states that characterise a feasible way in which recurrent execution of the task may occur. [1]

Pros

- Gives the right intuition
- Identifies that "scenario of operation" is the key notion

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

pWCET DEFINITION IS OPEN TO INTERPRETATION

► **Definition 2.** The *probabilistic Worst-Case Execution Time (pWCET) distribution* for a task is the least upper bound, in the sense of the greater than or equal to operator \succeq (defined below), on the execution time distribution of the jobs of the task for every valid *scenario of operation*, where a *scenario of operation* is defined as an infinitely repeating sequence of input states (including both input values and software state variables) and initial hardware states that characterise a feasible way in which recurrent execution of the task may occur. [1]

Pros

- Gives the right intuition
- Identifies that "scenario of operation" is the key notion

Cons

- Open to interpretation 🤔
- Key aspects are stated in **prose** only
- **Not** suitable for formal verification

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

pWCET DEFINITION IS OPEN TO INTERPRETATION

► **Definition 2.** The *probabilistic Worst-Case Execution Time (pWCET) distribution* for a task is the least upper bound, in the sense of the greater than or equal to operator \succeq (defined below), on the execution time distribution of the jobs of the task for every valid *scenario of operation*, where a *scenario of operation* is defined as an infinitely repeating sequence of input states (including both input values and software state variables) and initial hardware states that characterise a feasible way in which recurrent execution of the task may occur. [1]

Pros

- Gives the right intuition
- Identifies that "scenario of operation" is the key notion

Cons

- Open to interpretation 🤔
- Key aspects are stated in **prose** only
- **Not** suitable for formal verification
- **Does not** necessarily enable IID-based analyses

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

SOTA pWCET *DOES NOT* ENABLE IID ANALYSIS

Already noted in [1]

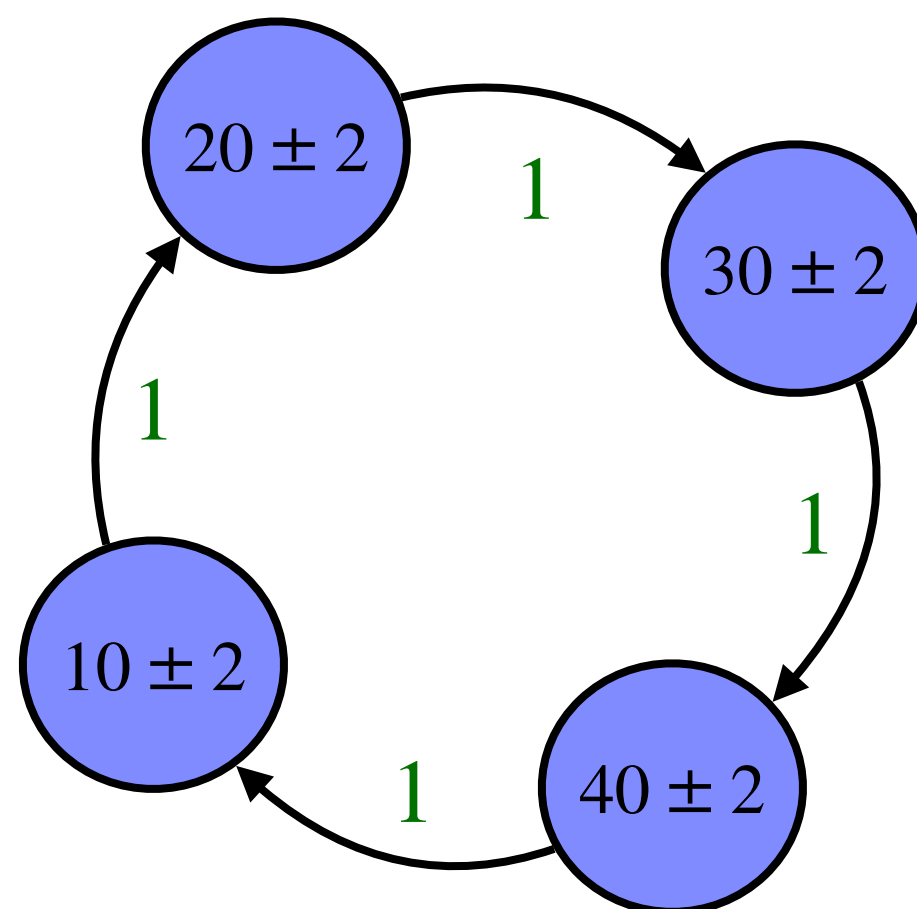
[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic timing analysis techniques for real-time systems."

SOTA pWCET *DOES NOT* ENABLE IID ANALYSIS

Already noted in [1]

A toy system: [1]

- Time-predictable hardware
- System has **four** states
- State **cycling through** its four possible values
- Small variability in each of the states
- Starts with random state



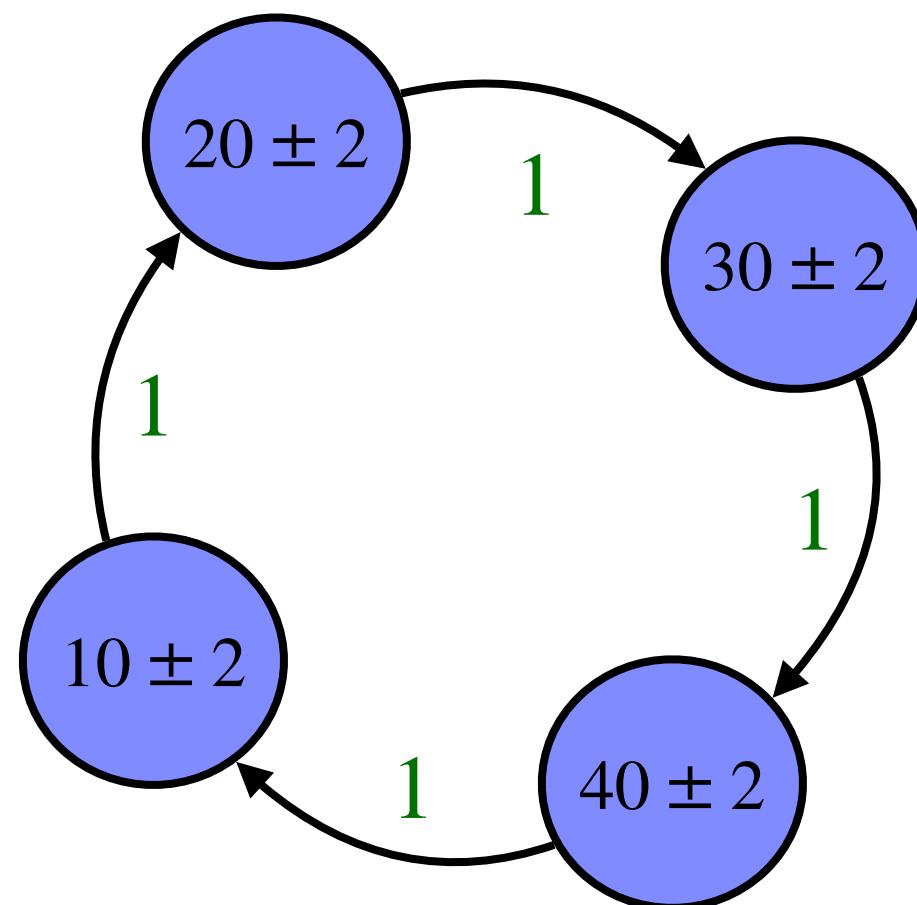
[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic timing analysis techniques for real-time systems."

SOTA pWCET *DOES NOT* ENABLE IID ANALYSIS

Already noted in [1]

A toy system: [1]

- Time-predictable hardware
- System has **four** states
- State **cycling through** its four possible values
- Small variability in each of the states
- Starts with random state



→ Resulting pET distribution: [1]

$$\begin{pmatrix} 10 \pm 2 & 20 \pm 2 & 30 \pm 2 & 40 \pm 2 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$$

→ Valid pWCET distribution: [1]

$$\begin{pmatrix} 12 & 22 & 32 & 42 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$$

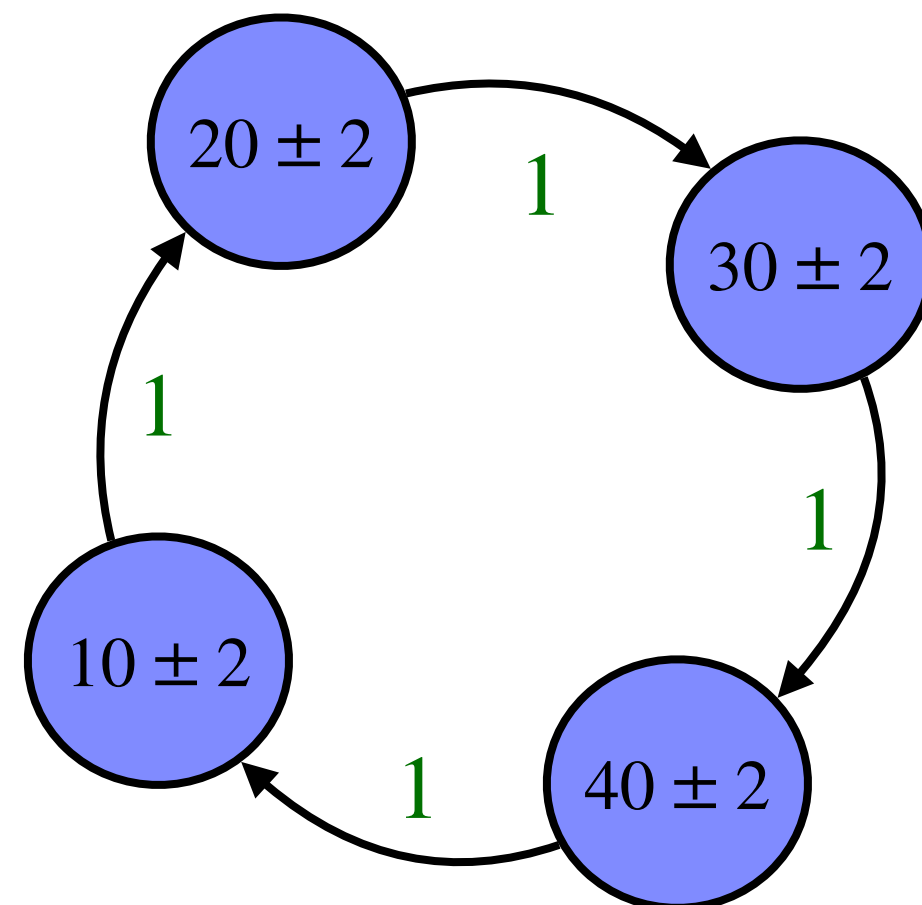
[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic timing analysis techniques for real-time systems."

SOTA pWCET *DOES NOT* ENABLE IID ANALYSIS

Already noted in [1]

A toy system: [1]

- Time-predictable hardware
- System has **four** states
- State **cycling through** its four possible values
- Small variability in each of the states
- Starts with random state



→ Resulting pET distribution: [1]

$$\begin{pmatrix} 10 \pm 2 & 20 \pm 2 & 30 \pm 2 & 40 \pm 2 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$$

→ Valid pWCET distribution: [1]

$$\begin{pmatrix} 12 & 22 & 32 & 42 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$$

Except that

→ Smallest workload of four consecutive jobs:

$$\rightarrow (10 - 2) + (20 - 2) + (30 - 2) + (40 - 2) = 92$$

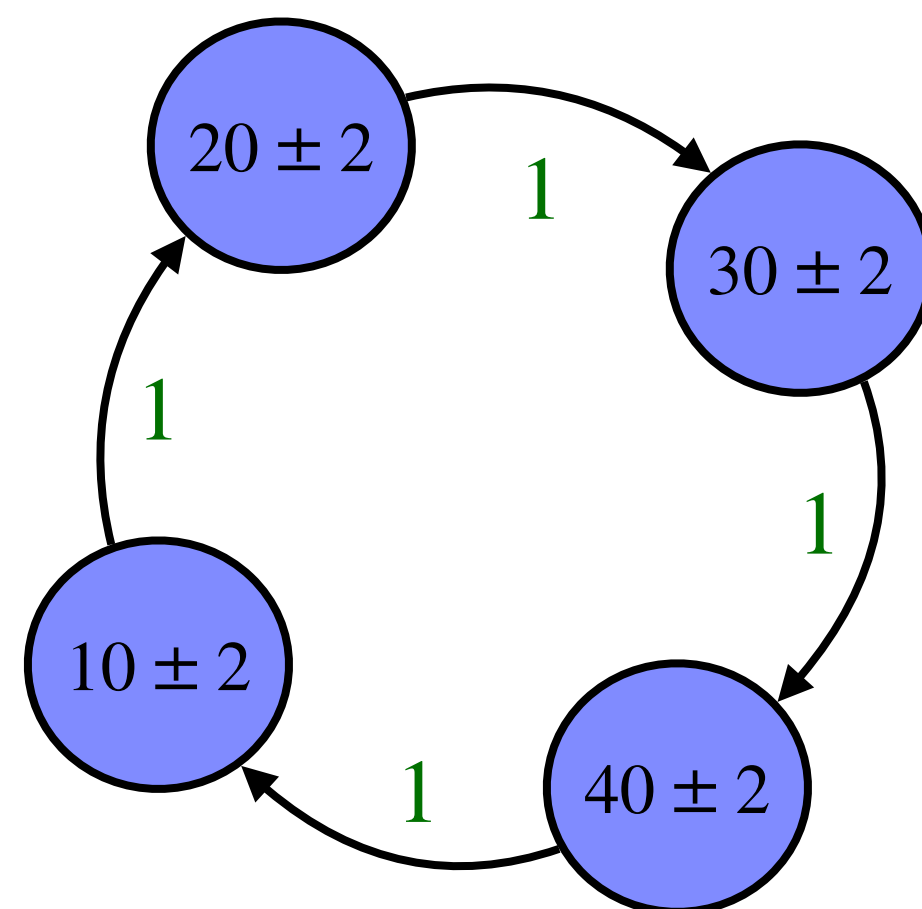
$$\mathbb{P} \left[\sum_4 pET \geq 92 \right] = 1$$

SOTA pWCET *DOES NOT* ENABLE IID ANALYSIS

Already noted in [1]

A toy system: [1]

- Time-predictable hardware
- System has **four** states
- State **cycling through** its four possible values
- Small variability in each of the states
- Starts with random state



→ Resulting pET distribution: [1]

$$\begin{pmatrix} 10 \pm 2 & 20 \pm 2 & 30 \pm 2 & 40 \pm 2 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$$

→ Valid pWCET distribution: [1]

$$\begin{pmatrix} 12 & 22 & 32 & 42 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$$

Except that

- Smallest workload of four consecutive jobs:
 - $(10 - 2) + (20 - 2) + (30 - 2) + (40 - 2) = 92$
- Sum of four pWCETs is insufficient:
 - E.g., $12 + 12 + 12 + 12 = 48$ has nonzero probability

$$\mathbb{P} \left[\sum_4 \text{pET} \geq 92 \right] = 1$$

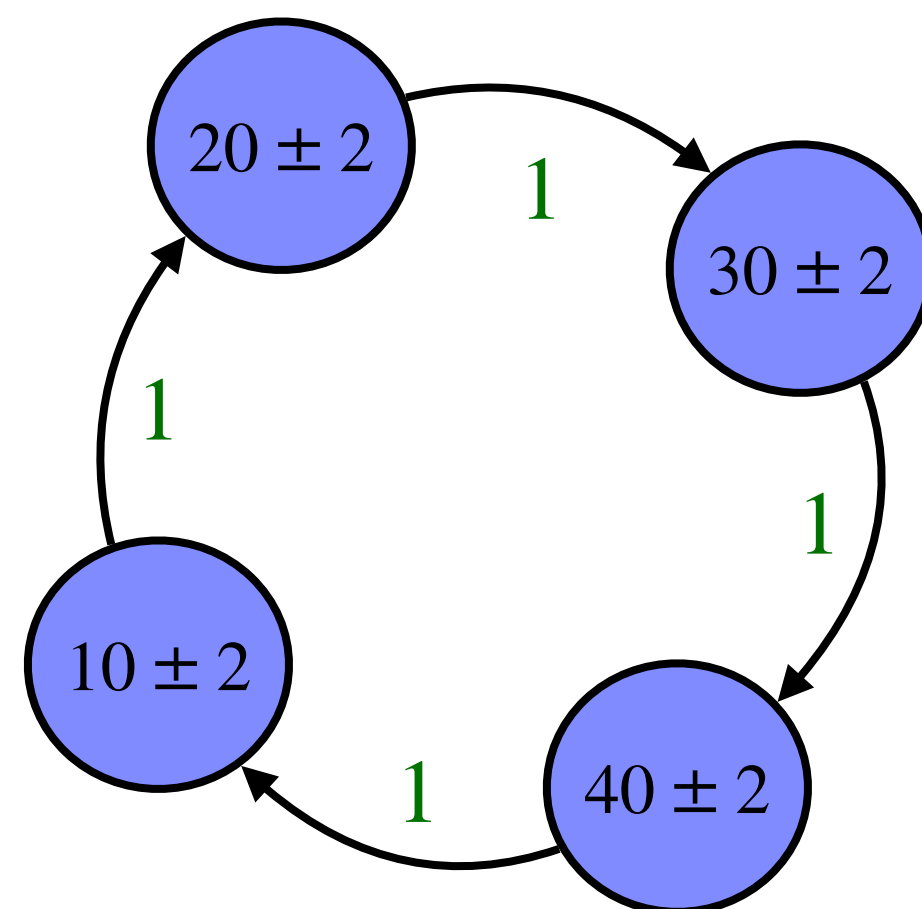
$$\mathbb{P} \left[\sum_4 \text{pWCET} \geq 92 \right] < 1$$

SOTA pWCET *DOES NOT* ENABLE IID ANALYSIS

Already noted in [1]

A toy system: [1]

- Time-predictable hardware
- System has **four** states
- State **cycling through** its four possible values
- Small variability in each of the states
- Starts with random state



→ Resulting pET distribution: [1]

$$\begin{pmatrix} 10 \pm 2 & 20 \pm 2 & 30 \pm 2 & 40 \pm 2 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$$

→ Valid pWCET distribution: [1]

$$\begin{pmatrix} 12 & 22 & 32 & 42 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$$



Not "appropriate" for IID-based analysis

Except that

- Smallest workload of four consecutive jobs:
 - $(10 - 2) + (20 - 2) + (30 - 2) + (40 - 2) = 92$
- Sum of four pWCETs is insufficient:
 - E.g., $12 + 12 + 12 + 12 = 48$ has nonzero probability

$$\mathbb{P} \left[\sum_4 pET \geq 92 \right] = 1$$

$$\mathbb{P} \left[\sum_4 pWCET \geq 92 \right] < 1$$

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic timing analysis techniques for real-time systems."

SOTA pWCET *DOES NOT* ENABLE IID ANALYSIS

Already noted in [1]

→ Resulting pET distribution: [1]

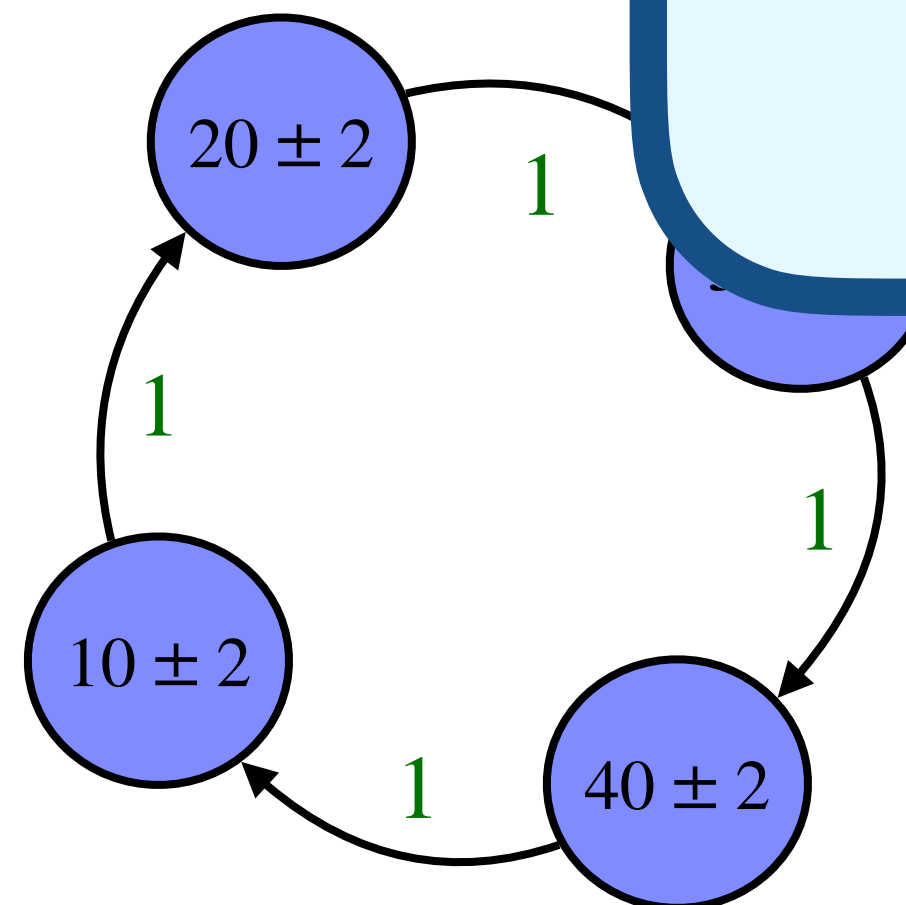
$$\left(\begin{array}{cccc} 10 \pm 2 & 20 \pm 2 & 30 \pm 2 & 40 \pm 2 \\ & & & 1/4 \end{array} \right)$$

A toy system: [1]

- Time-predictable hardware
- System has **four** components
- State **cycling** through the components
- Small variability in execution times
- Starts with random state

So, what is "appropriate" pWCET?

"appropriate" for IID-based analysis



→ $(10 - 2) + (20 - 2) + (30 - 2) + (40 - 2) = 92$

→ Sum of four pWCETs is insufficient:

→ E.g., $12 + 12 + 12 + 12 = 48$ has nonzero probability

$$\mathbb{P} \left[\sum_4 \text{pET} \geq 92 \right] = 1$$

$$\mathbb{P} \left[\sum_4 \text{pWCET} \geq 92 \right] < 1$$

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic timing analysis techniques for real-time systems."

OUR PROPOSAL:
AXIOMATIC pWCET

DESIGN GOALS

DESIGN GOALS

Formal definitions of pET and pWCET

- *Definitions that are mathematically formal and unambiguous*



DESIGN GOALS

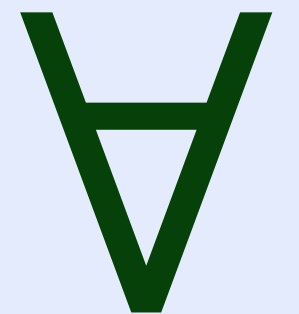
Formal definitions of pET and pWCET

- ▶ *Definitions that are mathematically formal and unambiguous*



Adequacy property: pWCET enables IID analysis

- ▶ *Any sound analysis assuming IID costs must result in a valid estimation*



DESIGN GOALS

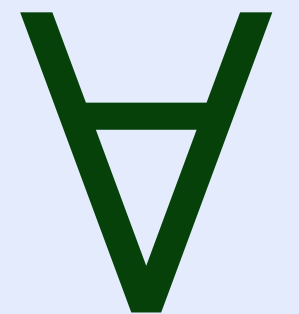
Formal definitions of pET and pWCET

- *Definitions that are mathematically formal and unambiguous*



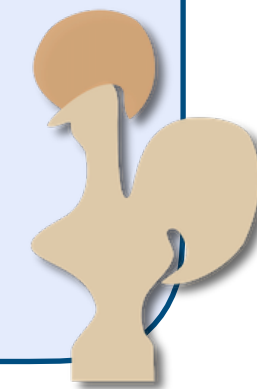
Adequacy property: pWCET enables IID analysis

- *Any sound analysis assuming IID costs must result in a valid estimation*



Precise enough to be **mechanisable** in Coq

- *pET, pWCET and adequacy property with its proof must be formalisable in Coq proof assistant*



AXIOMATIC pWCET

► **Definition 2.** The *probabilistic Worst-Case Execution Time (pWCET)* distribution for a task is the least upper bound, in the sense of the greater than or equal to operator \succeq (defined below), on the execution time distribution of the jobs of the task for every valid scenario of operation, where a *scenario* of operation is defined as an infinitely repeating sequence of input states (including both input values and software state variables) and initial hardware states that characterise a feasible way in which recurrent execution of the task may occur. [1]

AXIOMATIC pWCET

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

► **Definition 2.** The *probabilistic Worst-Case Execution Time (pWCET)* distribution for a task is the least upper bound, in the sense of the greater than or equal to operator \succeq (defined below), on the execution time distribution of the jobs of the task for every valid scenario of operation, where a *scenario* of operation is defined as an infinitely repeating sequence of input states (including both input values and software state variables) and initial hardware states that characterise a feasible way in which recurrent execution of the task may occur. [1]

AXIOMATIC pWCET

Def. 7 (♣).

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

► **Definition 2.** The *probabilistic Worst-Case Execution Time (pWCET) distribution* for a task is the least upper bound, in the sense of the greater than or equal to operator \succeq (defined below), on the execution time distribution of the jobs of the task for every valid scenario of operation, where a *scenario* of operation is defined as an infinitely repeating sequence of input states (including both input values and software state variables) and initial hardware states that characterise a feasible way in which recurrent execution of the task may occur. [1]

AXIOMATIC pWCET

Def. 7 (♣). A monotonically increasing function $F_i: \mathbb{W} \rightarrow [0, 1]$ with $F_i(0) = 0$ and $\lim_{t \rightarrow \infty} F_i(t) = 1$ is an axiomatic pWCET for a task τ_i if

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

► **Definition 2.** The *probabilistic Worst-Case Execution Time (pWCET) distribution* for a task is the least upper bound, in the sense of the greater than or equal to operator \succeq (defined below), on the execution time distribution of the jobs of the task for every valid scenario of operation, where a *scenario* of operation is defined as an infinitely repeating sequence of input states (including both input values and software state variables) and initial hardware states that characterise a feasible way in which recurrent execution of the task may occur. [1]

AXIOMATIC pWCET

Def. 7 (♣). A monotonically increasing function $F_i: \mathbb{W} \rightarrow [0, 1]$ with $F_i(0) = 0$ and $\lim_{t \rightarrow \infty} F_i(t) = 1$ is an axiomatic pWCET for a task τ_i if, for every $J \in \tau_i$ and every fixed arrival sequence $\xi \in \Xi$, there exists a partition \mathfrak{S} (Def. 4) such that

Def. 4 (♣). A partition $\mathfrak{S} \triangleq \{S_l\}_l$ is any finite, or countably infinite, disjoint cover of all positive-probability elements of Ω .

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

► **Definition 2.** The *probabilistic Worst-Case Execution Time (pWCET) distribution* for a task is the least upper bound, in the sense of the greater than or equal to operator \succeq (defined below), on the execution time distribution of the jobs of the task for every valid scenario of operation, where a *scenario* of operation is defined as an infinitely repeating sequence of input states (including both input values and software state variables) and initial hardware states that characterise a feasible way in which recurrent execution of the task may occur. [1]

AXIOMATIC pWCET

Def. 7 (P). A monotonically increasing function $F_i: \mathbb{W} \rightarrow [0, 1]$ with $F_i(0) = 0$ and $\lim_{t \rightarrow \infty} F_i(t) = 1$ is an axiomatic pWCET for a task τ_i if, for every $J \in \tau_i$ and every fixed arrival sequence $\xi \in \Xi$, there exists a partition \mathfrak{S} (Def. 4) such that

2) F_i \mathfrak{S} -dominates \mathcal{C}_J w.r.t. ξ (Def. 6).

Def. 4 (P). A partition $\mathfrak{S} \triangleq \{S_l\}_l$ is any finite, or countably infinite, disjoint cover of all positive-probability elements of Ω .

Def. 6 (P). Given a job $J \in \mathbb{J}$, a fixed arrival sequence ξ , and a partition \mathfrak{S} , a function $F: \mathbb{W} \rightarrow [0, 1]$ \mathfrak{S} -dominates \mathcal{C}_J iff

$$\forall S_l \in \mathfrak{S} \text{ s.th. } \mathbb{P}[S_l \wedge \xi] > 0: \mathbb{F}[\mathcal{C}_J | S_l \wedge \xi] \preceq F.$$

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

► **Definition 2.** The *probabilistic Worst-Case Execution Time (pWCET) distribution* for a task is the least upper bound, in the sense of the greater than or equal to operator \succeq (defined below), on the execution time distribution of the jobs of the task for every valid scenario of operation, where a *scenario* of operation is defined as an infinitely repeating sequence of input states (including both input values and software state variables) and initial hardware states that characterise a feasible way in which recurrent execution of the task may occur. [1]

AXIOMATIC pWCET

Def. 7 (P). A monotonically increasing function $F_i: \mathbb{W} \rightarrow [0, 1]$ with $F_i(0) = 0$ and $\lim_{t \rightarrow \infty} F_i(t) = 1$ is an axiomatic pWCET for a task τ_i if, for every $J \in \tau_i$ and every fixed arrival sequence $\xi \in \Xi$, there exists a partition \mathfrak{S} (Def. 4) such that

- 1) \mathcal{C}_J is partition-independent w.r.t. ξ and \mathfrak{S} (Def. 5), and
- 2) F_i \mathfrak{S} -dominates \mathcal{C}_J w.r.t. ξ (Def. 6).

Def. 5 (P). Given a job $J \in \mathbb{J}$, a fixed arrival sequence ξ , and a partition \mathfrak{S} , job J 's pET is partition-independent w.r.t. \mathfrak{S} iff, for any set $G \subseteq \mathbb{J}$ with $J \notin G$ and any fixed cost vector \vec{c}_\bullet :

$$\begin{aligned} \forall S_l \in \mathfrak{S} \text{ s.th. } \mathbb{P}[S_l \wedge \xi] > 0: \\ \mathbb{P}[\mathcal{C}_J = \vec{c}_J \wedge \forall J' \in G: \mathcal{C}_{J'} = \vec{c}_{J'} | S_l \wedge \xi] \\ = \mathbb{P}[\mathcal{C}_J = \vec{c}_J | S_l \wedge \xi] \cdot \mathbb{P}[\forall J' \in G: \mathcal{C}_{J'} = \vec{c}_{J'} | S_l \wedge \xi]. \end{aligned}$$

Def. 4 (P). A partition $\mathfrak{S} \triangleq \{S_l\}_l$ is any finite, or countably infinite, disjoint cover of all positive-probability elements of Ω .

Def. 6 (P). Given a job $J \in \mathbb{J}$, a fixed arrival sequence ξ , and a partition \mathfrak{S} , a function $F: \mathbb{W} \rightarrow [0, 1]$ \mathfrak{S} -dominates \mathcal{C}_J iff

$$\forall S_l \in \mathfrak{S} \text{ s.th. } \mathbb{P}[S_l \wedge \xi] > 0: \mathbb{F}[\mathcal{C}_J | S_l \wedge \xi] \preceq F.$$

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

► **Definition 2.** The *probabilistic Worst-Case Execution Time (pWCET) distribution* for a task is the least upper bound, in the sense of the greater than or equal to operator \succeq (defined below), on the execution time distribution of the jobs of the task for every valid scenario of operation, where a *scenario* of operation is defined as an infinitely repeating sequence of input states (including both input values and software state variables) and initial hardware states that characterise a feasible way in which recurrent execution of the task may occur. [1]

AXIOMATIC pWCET

Def. 7 (P). A monotonically increasing function $F_i: \mathbb{W} \rightarrow [0, 1]$ with $F_i(0) = 0$ and $\lim_{t \rightarrow \infty} F_i(t) = 1$ is an axiomatic pWCET for a task τ_i if, for every $J \in \tau_i$ and every fixed arrival sequence $\xi \in \Xi$, there exists a partition \mathfrak{S} (Def. 4) such that

- 1) \mathcal{C}_J is partition-independent w.r.t. ξ and \mathfrak{S} (Def. 5), and
- 2) F_i \mathfrak{S} -dominates \mathcal{C}_J w.r.t. ξ (Def. 6).

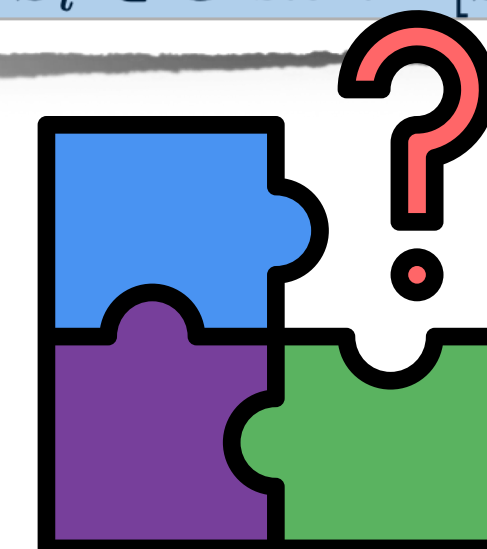
Def. 5 (P). Given a job $J \in \mathbb{J}$, a fixed arrival sequence ξ , and a partition \mathfrak{S} , job J 's pET is partition-independent w.r.t. \mathfrak{S} iff, for any set $G \subseteq \mathbb{J}$ with $J \notin G$ and any fixed cost vector \vec{c}_\bullet :

$$\begin{aligned} \forall S_l \in \mathfrak{S} \text{ s.th. } \mathbb{P}[S_l \wedge \xi] > 0: \\ \mathbb{P}[\mathcal{C}_J = \vec{c}_J \wedge \forall J' \in G: \mathcal{C}_{J'} = \vec{c}_{J'} | S_l \wedge \xi] \\ = \mathbb{P}[\mathcal{C}_J = \vec{c}_J | S_l \wedge \xi] \cdot \mathbb{P}[\forall J' \in G: \mathcal{C}_{J'} = \vec{c}_{J'} | S_l \wedge \xi]. \end{aligned}$$

Def. 4 (P). A partition $\mathfrak{S} \triangleq \{S_l\}_l$ is any finite, or countably infinite, disjoint cover of all positive-probability elements of Ω .

Def. 6 (P). Given a job $J \in \mathbb{J}$, a fixed arrival sequence ξ , and a partition \mathfrak{S} , a function $F: \mathbb{W} \rightarrow [0, 1]$ \mathfrak{S} -dominates \mathcal{C}_J iff

$$\forall S_l \in \mathfrak{S} \text{ s.th. } \mathbb{P}[S_l \wedge \xi] > 0: \mathbb{F}[\mathcal{C}_J | S_l \wedge \xi] \preceq F.$$



"Appropriate" pWCET part?

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

► **Definition 2.** The *probabilistic Worst-Case Execution Time (pWCET) distribution* for a task is the least upper bound, in the sense of the greater than or equal to operator \succeq (defined below), on the execution time distribution of the jobs of the task for every valid scenario of operation, where a *scenario* of operation is defined as an infinitely repeating sequence of input states (including both input values and software state variables) and initial hardware states that characterise a feasible way in which recurrent execution of the task may occur. [1]

AXIOMATIC pWCET

Def. 7 (P). A monotonically increasing function $F_i: \mathbb{W} \rightarrow [0, 1]$ with $F_i(0) = 0$ and $\lim_{t \rightarrow \infty} F_i(t) = 1$ is an axiomatic pWCET for a task τ_i if, for every $J \in \tau_i$ and every fixed arrival sequence $\xi \in \Xi$, there exists a partition \mathfrak{S} (Def. 4) such that

- 1) \mathcal{C}_J is partition-independent w.r.t. ξ and \mathfrak{S} (Def. 5), and
- 2) F_i \mathfrak{S} -dominates \mathcal{C}_J w.r.t. ξ (Def. 6).

Def. 4 (P). A partition $\mathfrak{S} \triangleq \{S_l\}_l$ is any finite, or countably infinite, disjoint cover of all positive-probability elements of Ω .

Def. 6 (P). Given a job $J \in \mathbb{J}$, a fixed arrival sequence ξ , and a partition \mathfrak{S} , a function $F: \mathbb{W} \rightarrow [0, 1]$ \mathfrak{S} -dominates \mathcal{C}_J iff

$$\forall S_l \in \mathfrak{S} \text{ s.th. } \mathbb{P}[S_l \wedge \xi] > 0: \mathbb{F}[\mathcal{C}_J | S_l \wedge \xi] \preceq F.$$

Def. 5 (P). Given a job $J \in \mathbb{J}$, a fixed arrival sequence ξ , and a partition \mathfrak{S} , job J 's pET is partition-independent w.r.t. \mathfrak{S} iff, for any set $G \subseteq \mathbb{J}$ with $J \notin G$ and any fixed cost vector \vec{c}_\bullet :

$$\begin{aligned} &\forall S_l \in \mathfrak{S} \text{ s.th. } \mathbb{P}[S_l \wedge \xi] > 0: \\ &\mathbb{P}[\mathcal{C}_J = \vec{c}_J \wedge \forall J' \in G: \mathcal{C}_{J'} = \vec{c}_{J'} | S_l \wedge \xi] \\ &= \mathbb{P}[\mathcal{C}_J = \vec{c}_J | S_l \wedge \xi] \cdot \mathbb{P}[\forall J' \in G: \mathcal{C}_{J'} = \vec{c}_{J'} | S_l \wedge \xi]. \end{aligned}$$

Axiomatic pWCET: scenario of operation must ensure that jobs' pETs become independent

[1] Davis, Robert Ian, and Liliana Cucu-Grosjean. "A survey of probabilistic schedulability analysis techniques for real-time systems."

ADEQUACY

ADEQUACY: FORMAL BASIS FOR IID REASONING

How do we know that an IID-based analysis that uses axiomatic pWCET will obtain a sound bound?

ADEQUACY: FORMAL BASIS FOR IID REASONING

How do we know that an IID-based analysis that uses axiomatic pWCET will obtain a sound bound?

Probabilistic
Response Time (pRT)

Intuitively, we want to prove:

→ Ground-truth pRT is \leq -bounded
by pRT derived via pWCETs

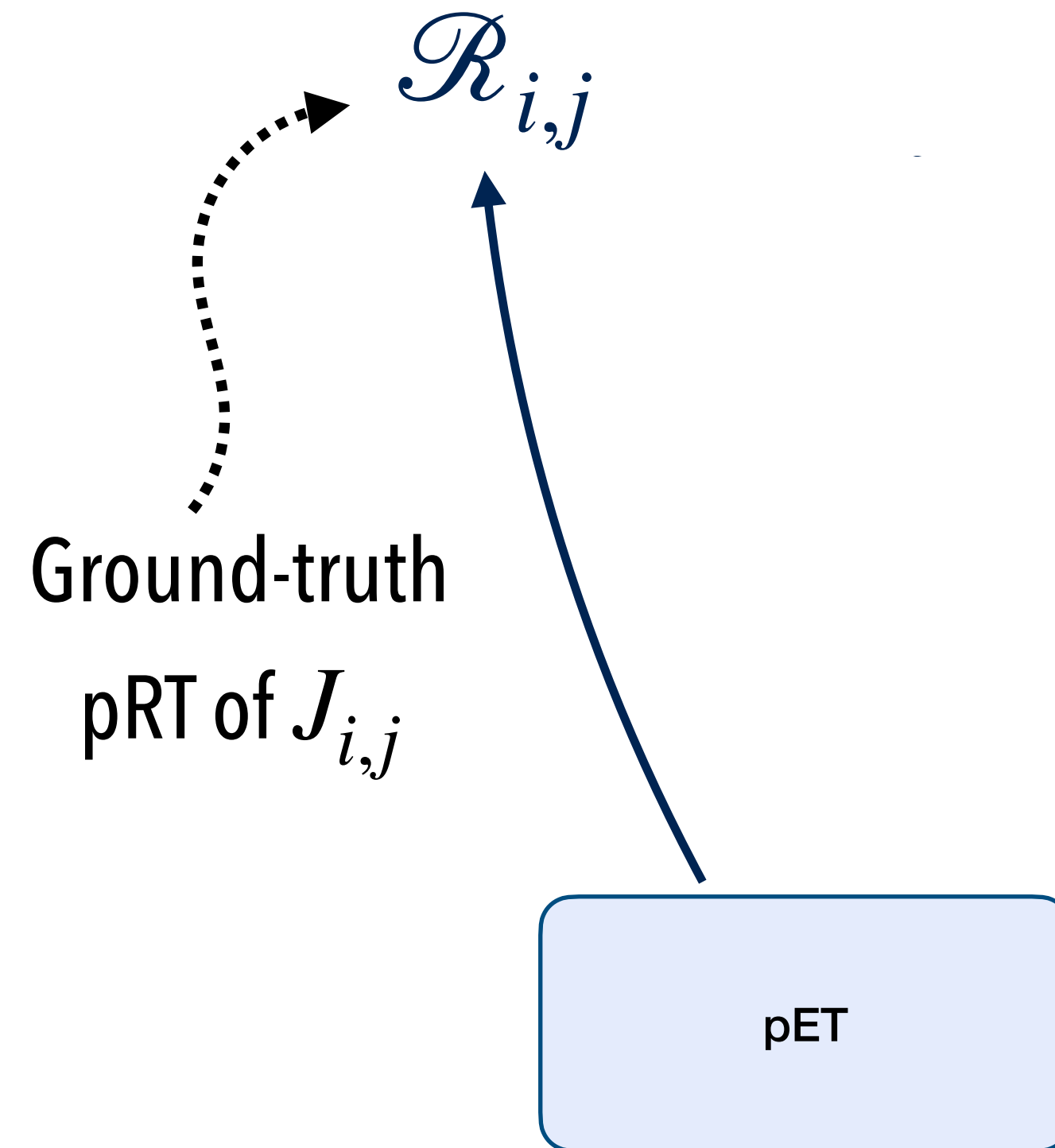
ADEQUACY: FORMAL BASIS FOR IID REASONING

How do we know that an IID-based analysis that uses axiomatic pWCET will obtain a sound bound?

Probabilistic
Response Time (pRT)

Intuitively, we want to prove:

→ Ground-truth pRT is \leq -bounded
by pRT derived via pWCETs



ADEQUACY: FORMAL BASIS FOR IID REASONING

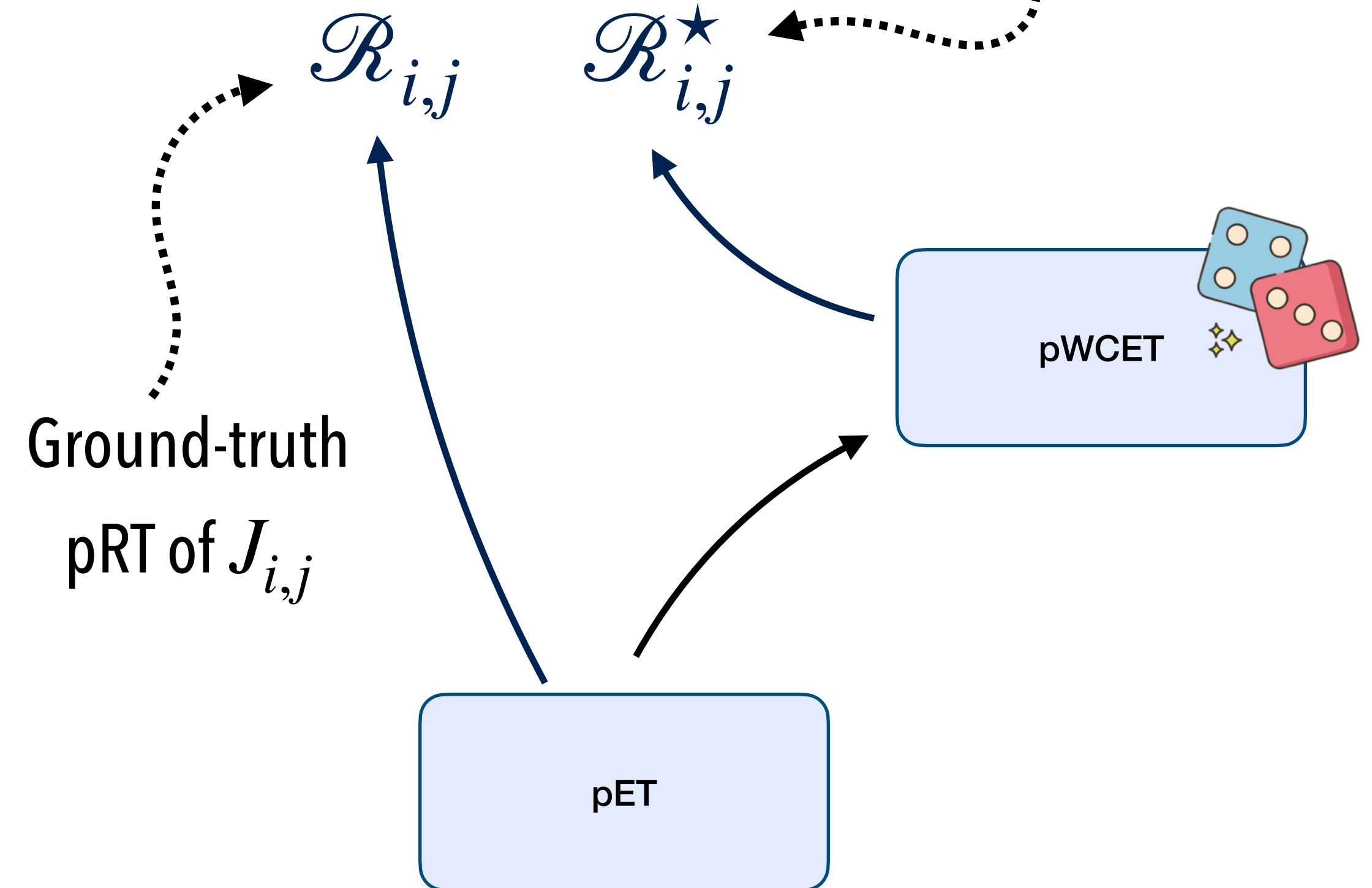
How do we know that an IID-based analysis that uses axiomatic pWCET will obtain a sound bound?

pRT of $J_{i,j}$ obtained by **any valid IID-based analysis** using axiomatic pWCET

Probabilistic
Response Time (pRT)

Intuitively, we want to prove:

→ Ground-truth pRT is \leq -bounded by pRT derived via pWCETs



ADEQUACY: FORMAL BASIS FOR IID REASONING

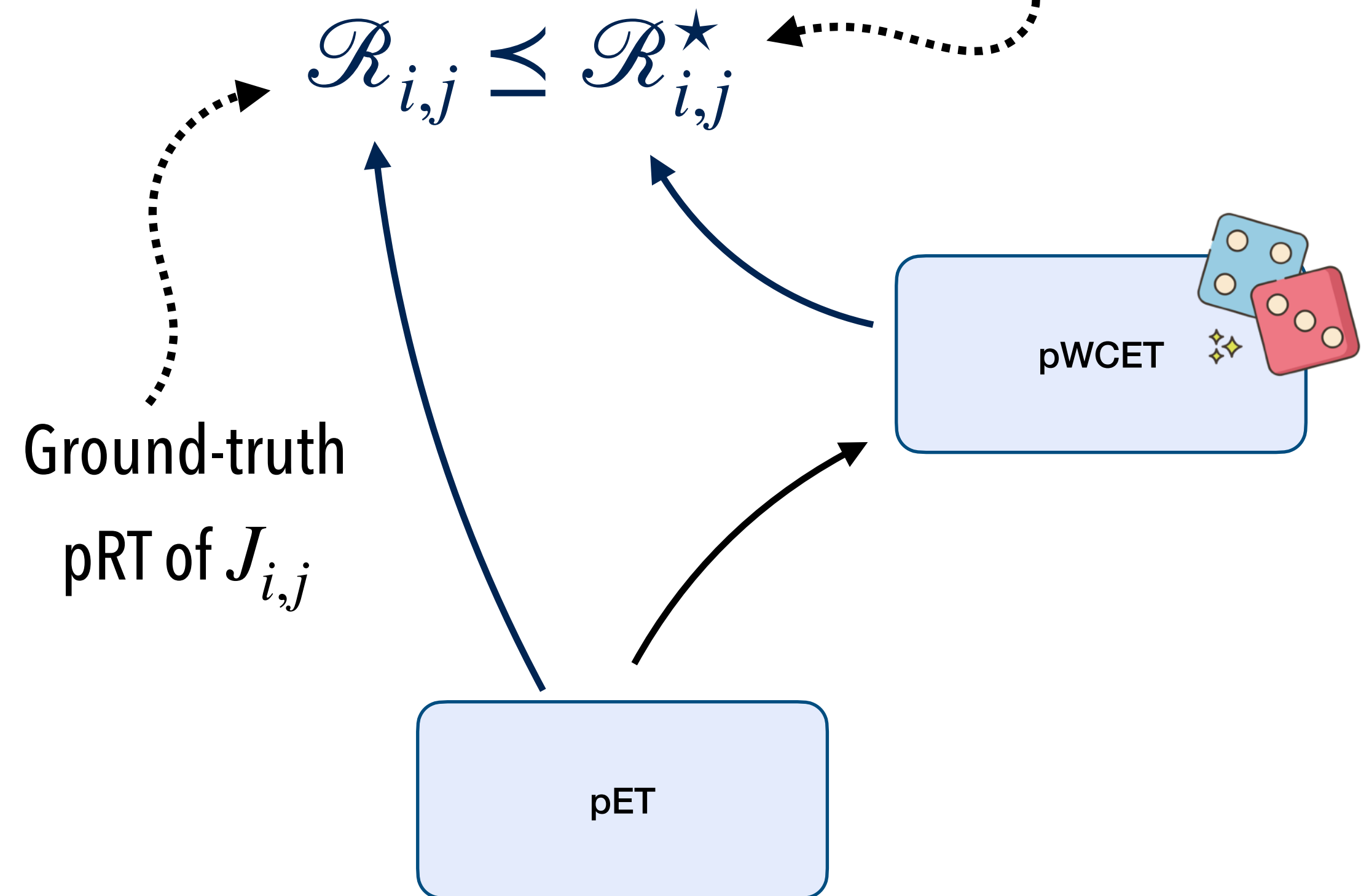
How do we know that an IID-based analysis that uses axiomatic pWCET will obtain a sound bound?

pRT of $J_{i,j}$ obtained by **any valid IID-based analysis** using axiomatic pWCET

Probabilistic Response Time (pRT)

Intuitively, we want to prove:

→ Ground-truth pRT is \leq -bounded by pRT derived via pWCETs



ADEQUACY: FORMAL BASIS FOR IID REASONING

How do we know that an IID-based analysis that uses axiomatic pWCET will obtain a sound bound?

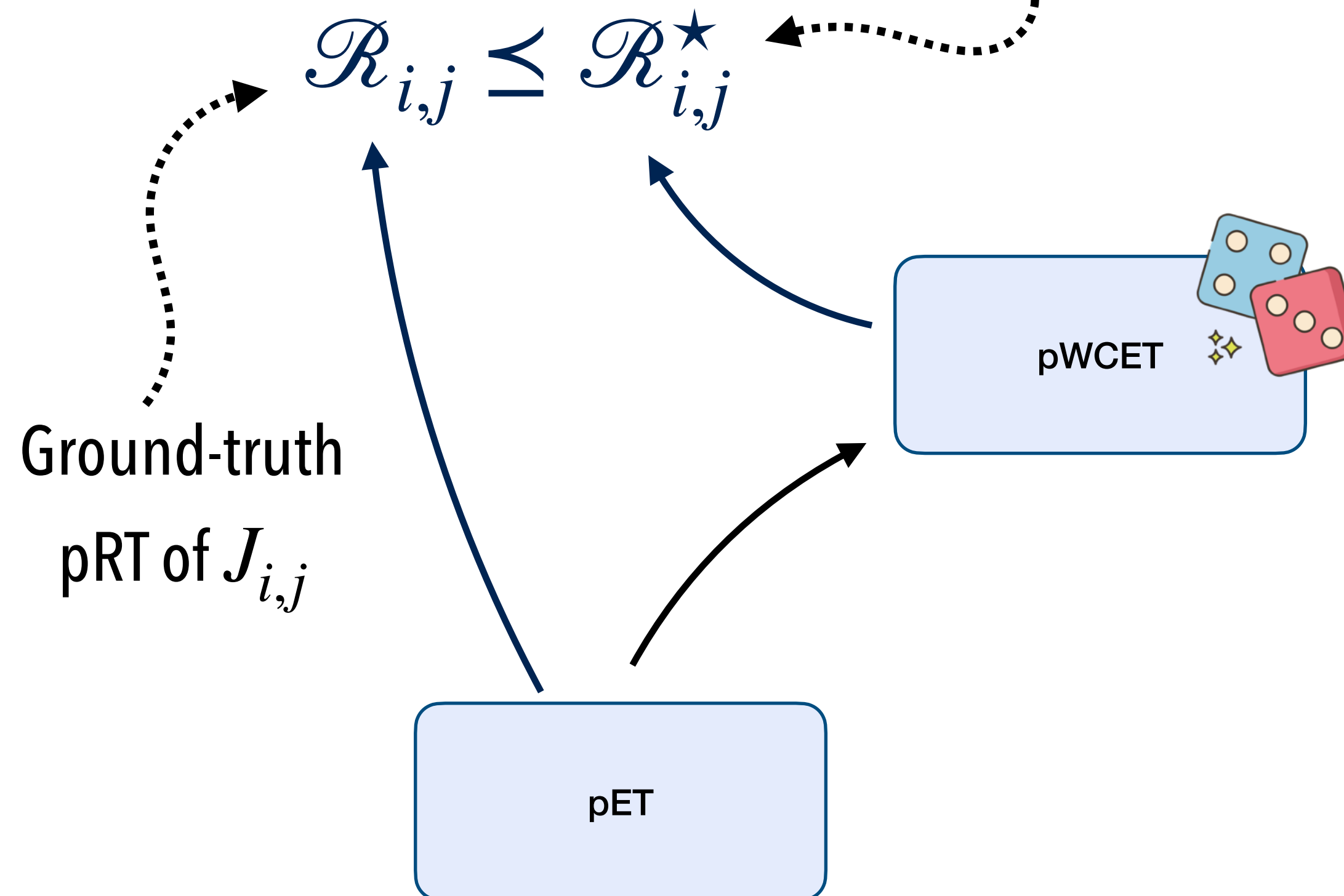
pRT of $J_{i,j}$ obtained by **any valid IID-based analysis** using axiomatic pWCET

Probabilistic Response Time (pRT)

Intuitively, we want to prove:

→ Ground-truth pRT is \leq -bounded by pRT derived via pWCETs

Formal statement is **surprisingly** tricky and involves the notion of "replacement" of pETs with pWCETs



AXIOMATIC pWCET IS ADEQUATE

Theorem (*paraphrased*). Consider a job $J_{i,j}$. Let $\mathcal{R}_{i,j}$ be the pRT of $J_{i,j}$ in the initial system and $\mathcal{R}_{i,j}^\star$ be the pRT of $J_{i,j}$ in a simplified system obtained via pWCET F_i . Then $\mathcal{R}_{i,j} \leq \mathcal{R}_{i,j}^\star$.

AXIOMATIC pWCET IS ADEQUATE

Theorem (*paraphrased*). Consider a job $J_{i,j}$. Let $\mathcal{R}_{i,j}$ be the pRT of $J_{i,j}$ in the initial system and $\mathcal{R}_{i,j}^\star$ be the pRT of $J_{i,j}$ in a simplified system obtained via pWCET F_i . Then $\mathcal{R}_{i,j} \leq \mathcal{R}_{i,j}^\star$.

Hint:

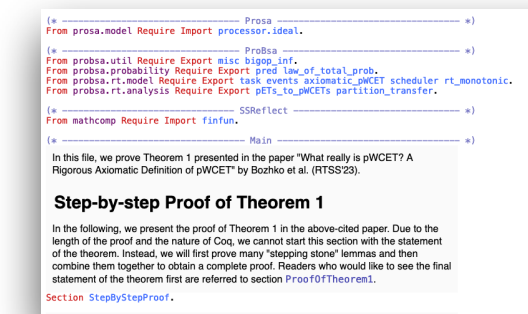
1. Use axiomatic pWCET to construct a "copy" of the initial system, where pETs are replaced with job costs that are, **by construction**, IID and have distribution F_i
2. Prove that pRT $\mathcal{R}_{i,j}^\star$ in the simplified system stochastically dominates the original pRT $\mathcal{R}_{i,j}$

AXIOMATIC pWCET IS ADEQUATE

Theorem (*paraphrased*). Consider a job $J_{i,j}$. Let $\mathcal{R}_{i,j}$ be the pRT of $J_{i,j}$ in the initial system and $\mathcal{R}_{i,j}^\star$ be the pRT of $J_{i,j}$ in a simplified system obtained via pWCET F_i . Then $\mathcal{R}_{i,j} \leq \mathcal{R}_{i,j}^\star$.

Hint:

1. Use axiomatic pWCET to construct a "copy" of the initial system, where pETs are replaced with job costs that are, **by construction**, IID and have distribution F_i
2. Prove that pRT $\mathcal{R}_{i,j}^\star$ in the simplified system stochastically dominates the original pRT $\mathcal{R}_{i,j}$



AXIOMATIC pWCET IS ADEQUATE

Theorem (*paraphrased*). Consider a job $J_{i,j}$. Let $\mathcal{R}_{i,j}$ be the pRT of $J_{i,j}$ in the initial system and $\mathcal{R}_{i,j}^\star$ be the pRT of $J_{i,j}$ in a **simplified system obtained via pWCET F_i** . Then $\mathcal{R}_{i,j} \leq \mathcal{R}_{i,j}^\star$.

Hint:

1. Use axiomatic pWCET to construct a "copy" of the initial system, where pETs are replaced with job costs that are, **by construction**, IID and have distribution F_i
2. Prove that pRT $\mathcal{R}_{i,j}^\star$ in the simplified system stochastically dominates the original pRT $\mathcal{R}_{i,j}$

```

From proba.model Require Export processor_label.
From proba.util Require Export basic_prob_util.
From proba.probability Require Export prob_util_of_set_of_prob.
From proba.ft.model Require Export task_events_schedule_pWCET_scheduler_ft_monotonic.
From proba.ft.analysis Require Export pETs_to_pWCET_partition_transfer.
From mathcomp Require Import flatmap.

Section Proof.
  In this file, we prove Theorem 1 presented in the paper "What really is pWCET? A Rigorous Axiomatic Definition of pWCET" by Bozhko et al. (ITDSS23).
  *Step-by-step Proof of Theorem 1
  In the following, we present the proof of Theorem 1 in the above-cited paper. Due to the length of the proof and the nature of Coq, we cannot start this section with the statement of the theorem. Instead, we will first prove many "stepping stone" lemmas and then combine them together to obtain a complete proof. Readers who would like to see the final statement of the theorem first are referred to section ProofOfTheorem1.
  End Section.

```

Step-by-step Proof of Theorem 1

In the following, we present the proof of Theorem 1 in the above-cited paper. Due to the length of the proof and the nature of Coq, we cannot start this section with the statement of the theorem. Instead, we will first prove many "stepping stone" lemmas and then combine them together to obtain a complete proof. Readers who would like to see the final statement of the theorem first are referred to section [ProofOfTheorem1](#).

AXIOMATIC pWCET IS ADEQUATE

Theorem (paraphrased). Consider a job $J_{i,j}$. Let $\mathcal{R}_{i,j}$ be the pRT of $J_{i,j}$ in the initial system and $\mathcal{R}_{i,j}^\star$ be the pRT of $J_{i,j}$ in a simplified system obtained via pWCET F_i . Then $\mathcal{R}_{i,j} \leq \mathcal{R}_{i,j}^\star$.

Hint:

1. Use axiomatic pWCET to construct a "copy" of the initial system, where pETs are replaced with job costs that are, **by construction**, IID and have distribution F_i
2. Prove that pRT $\mathcal{R}_{i,j}^\star$ in the simplified system stochastically dominates the original pRT $\mathcal{R}_{i,j}$

```

From probas.model Require Export processor_model.
From probas.model Require Export task_model.
From probas.model Require Export task_model_partition_transfer.
From probas.model Require Export pWCET_partition_transfer.
From probas.analysis Require Export pWCET_partition_transfer.
From probas.analysis Require Export flatmap.
From natcomp Require Import flatmap.

Section StepByStepProof.
  In this file, we prove Theorem 1 presented in the paper "What really is pWCET? A Rigorous Axiomatic Definition of pWCET" by Bozhko et al. (RTSS22).

  Section StepByStepProof.
  In this section, we demonstrate a step-by-step proof of the main claim (Theorem 1 from the paper). It is important to note that Coq's preferred method of proof is a "bottom-up" approach, while most humans are better following a top-down derivation. In other words, in Coq it is easier to present proofs in the following way: first simple facts are proven, then one can use these simple facts to construct more sophisticated facts, then even more sophisticated ones, and so on until the final goal is reached. Yet, a reasonable question for a human would be "where is the goal?"

  Therefore, we adopt a more paper-like approach in which we present the proof as a series of implications: C  $\Rightarrow$  Goal, B  $\Rightarrow$  C, A  $\Rightarrow$  B. This allows us to begin with the final goal (assuming that some facts are given) and then gradually work our way towards the "nuts" of the overall argument pertaining to specific details.

  Assume horizon defines an upper bound on the termination time of the system. If horizon shows, the system does not necessarily terminate. Note, however, that in either case our proof assumes there to be a finite number of jobs for technical reasons. As the horizon can be chosen to be arbitrarily large (the proof does not depend on its magnitude), e.g., hundreds of even thousands of years, assuming the existence of a finite horizon is not unreasonable for a computing system.

  Variable horizon : option nat.
  Context {Task : TaskType}
  Context {Task : TaskType}
  Context {Task : TaskType}.

  ...and their jobs.
  Context {Job : JobType}
  Context {Job : JobType}
  Context {Job : JobType}.

  Consider a response time monotonic scheduling algorithm C, where response time monotonic means the following: assuming that all arrival times are fixed, an increase of the execution cost of any job cannot cause a decrease in the response time of any job. Recall that C receives two vectors: a vector of arrival times A and a vector of job costs C.
  Variable C : SchedulingTask Job.
  Hypothesis H_monotonic : RT_monotonic_scheduler horizon C.

  For brevity, let sched denote the probabilistic schedule generated by C for a given system S.
  Let sched S := compute_scheduler C (job_arrival := A_of S) (job_cost := C_of S).

  As before, consider four parameters that describe a system under analysis: a sample space  $\Omega$ , a measure  $\mu$ , job arrival times  $a_j$ , and job execution costs  $c_j$ .
  Variable  $\Omega$  : Type.
  Variable  $\mu$  : Measure  $\Omega$ .
  Variable  $a$  : SchedulingTask Job  $\Omega$ .
  Variable  $c$  : SchedulingTask Job  $\Omega$ .

  Let us use these parameters to construct a system S.
  Let S := { |  $\Omega$ ,  $\mu$ ,  $a$ ,  $c$  |}.

  Next, we assume that the aforementioned pWCET is an axiomatic pWCET. That is, for any job j and any arrival sequence C, there exists a partition  $\pi$  of job positive probability events such that both partition dominance and partition independence are satisfied.
  Hypothesis H_axiomatic_pWCET :
  axiomatic_pWCET (A_of S) (job_arrival := A_of S) (job_cost := C_of S).

  Suppose we use the construction replace_job_pWCET presented in probas/rt_monotonic/replace_job_pWCET to replace the execution cost of given job j. Let S' denote the resulting system.
  Variable j : Job.
  Let S' := replace_job_pWCET j _rep S.

  For convenience, let task denote j's task.
  Let task := job_task j _rep.

  ... and let task denote the measure induced by pWCET.
  Let task :=
  match pWCET j _rep with
  | H1 |> pWCET task _rep :=
  (| task := H1 |> pWCET task _rep)
  end.

  Next, consider an arbitrary job j of any task and a duration r.
  Variables (j : Job) (r : duration).

  Finally, let Exc denote the event that j's response time exceeds r time units in system S.
  Let Exc :=  $\lambda w \in \Omega$ , S, exceeds (response_time (job_arrival := A_of S) (job_cost := C_of S) (sched S) horizon) w).

  ... and let Exc' denote the event that j's response time exceeds r time units in system S'.
  Let Exc' :=  $\lambda w \in \Omega$ , S', exceeds (response_time (job_arrival := A_of S') (job_cost := C_of S') (sched S') horizon) w).

  The remainder of this file serves for the most part to relate the probability of Exc with the probability of Exc', namely to establish that  $P_{\Omega, \mu, a, c}[\text{Exc}] = P_{\Omega, \mu, a, c}[\text{Exc}']$ .

  Step 1
  Now we are ready to start the proof. First, we do a case analysis on all possible arrival sequences.
  Section Step1.
  Consider a partition  $\pi$  of the sample space  $\Omega$  of S into events corresponding to different arrival sequences.
  Let  $\pi$  := partition_on  $\pi$ .

  Here  $\pi$  is an (indexed) set of events, where an  $i$ -th event denoted as  $\pi[i]$  represents a subset of  $\Omega$  where the arrival sequence is equal to  $i$ .
  (Readers focused on technical details may be interested in noting that the indices of partition  $\pi$  are arrival sequences themselves. This detail may be safely skipped over by most casual readers.)

  As discussed in the paper, one can transfer the partition  $\pi$  to the system S' and denote it as  $\pi'$ . For further details see the function partition_transfer_extending_partition.
  Let  $\pi'$  := extend_partition S  $\pi$ .

  Partitions  $\pi$  and  $\pi'$  are so similar that one can prove equivalence. For example, two indices  $i$  and  $i'$  that are "trickle equivalent" can be shown to be identical. (Trickle equivalence is a very strong notion of equivalence: one can intuitively view it simply as an equality between two elements of similar type; for further details see https://doi.org/10.1017/etcs.2022.1)
  Remark (E1_E1') :
  "If  $i \in \pi$  and  $i' \in \pi'$ ,
  pick (a1)  $i \in \pi$  and  $i' \in \pi'$ ".

  Now, recall that we present the proof in a top-down (C  $\Rightarrow$  Goal) fashion, starting with the overall theorem.
  So, assuming that for any two elements of partitions  $\pi$  and  $\pi'$  that are "trickle equivalent", it holds that  $P[\text{Exc} \mid \text{Exc} \in \pi[i]] = P[\text{Exc}' \mid \text{Exc}' \in \pi'[i']]$ .

  Hypothesis H_inj :
  "If  $i \in \pi$  and  $i' \in \pi'$ ,
  pick (a1)  $i \in \pi$  and  $i' \in \pi'$ ,
  pick (a2)  $i' \in \pi'$  and  $i \in \pi$ ".

  ... we can show that  $P_{\Omega, \mu, a, c}[\text{Exc}] = P_{\Omega, \mu, a, c}[\text{Exc}']$ . Or, in other words, we reduced the lemma statement to the hypothesis statement.
  Lemma (transfer_probabilities_monotonic_sched) :
  "P_{\Omega, \mu, a, c}[\text{Exc}] = P_{\Omega, \mu, a, c}[\text{Exc}']".
  end Step1.

  Step 2
  Now, we know that if we have the inequality with partitions on arrival sequences  $\pi$  and  $\pi'$ , then we are done. But how do we prove such an inequality? In the next step, we prove the required inequality by introducing a new hypothesis from which it can be established. We then continue in this manner until reaching a hypothesis that is easy enough to prove without introducing new hypotheses.
  
```

AXIOMATIC pWCET IS ADEQUATE

Theorem (paraphrased). Consider a job $J_{i,j}$. Let $\mathcal{R}_{i,j}$ be the pRT of $J_{i,j}$ in the initial system and $\mathcal{R}_{i,j}^\star$ be the pRT of $J_{i,j}$ in a simplified system obtained via pWCET F_j . Then $\mathcal{R}_{i,j} \leq \mathcal{R}_{i,j}^\star$.

Hint:

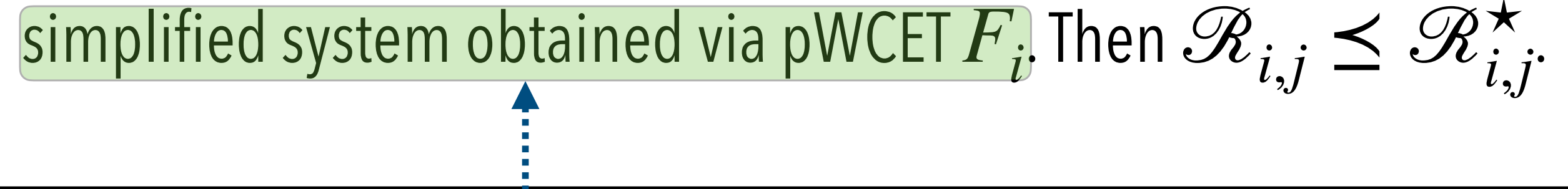
1. Use axiomatic pWCET to construct a "copy" of the initial system, where pETs are replaced with job costs that are, **by construction, IID** and have distribution F_i
2. Prove that pRT $\mathcal{R}_{i,j}^\star$ in the simplified system stochastically dominates the original pRT $\mathcal{R}_{i,j}$

The image shows a detailed technical document titled "Step-by-step Proof of Theorem 1". It contains several sections:

- Section Step 1:** Discusses the proof of Theorem 1, mentioning the need to prove the inequality with partitions on arrival sequences. It includes a lemma about the probability of events occurring within a certain time interval.
- Section Step 2:** Focuses on the partitioning of the sample space. It defines a partition \mathcal{E} and shows how to extend it to a new partition \mathcal{E}' . It includes a lemma about the probability of events occurring within a certain time interval.
- Section Step 3:** Discusses the construction of a simplified system. It defines a family of predicates part_i and shows how to use them to partition the sample space. It includes a lemma about the probability of events occurring within a certain time interval.
- Section Step 4:** Discusses the construction of a simplified system. It defines a family of predicates part_i and shows how to use them to partition the sample space. It includes a lemma about the probability of events occurring within a certain time interval.
- Section Step 5:** Discusses the construction of a simplified system. It defines a family of predicates part_i and shows how to use them to partition the sample space. It includes a lemma about the probability of events occurring within a certain time interval.

AXIOMATIC pWCET IS ADEQUATE

Theorem (paraphrased). Consider a job $J_{i,j}$. Let $\mathcal{R}_{i,j}$ be the pRT of $J_{i,j}$ in the initial system and $\mathcal{R}_{i,j}^\star$ be the pRT of $J_{i,j}$ in a simplified system obtained via pWCET F_j . Then $\mathcal{R}_{i,j} \leq \mathcal{R}_{i,j}^\star$.



Hint:

1. Use axiomatic pWCET to construct a "copy" of the initial system, where pETs are replaced with job costs that are, by construction, IID and have distribution F_i
2. Prove that pRT $\mathcal{R}_{i,j}^\star$ in the simplified system stochastically dominates the original pRT $\mathcal{R}_{i,j}$

```

...
pWCET_Sched := proc (J, F)
  local J, F;
  require J >= 0;
  require F >= 0;
  ...
end proc;

...

```

Step-by-step Proof of Theorem 1

In the following, we present the proof of Theorem 1 in the above-cited paper. Due to the length of the proof and the nature of Coq, we cannot start this section with the statement of the theorem. Instead, we will prove many "helping lemmas" and then combine them together to obtain a complete proof. Readers who would like to see the final statement of the theorem first are referred to section Proof/Theorem.

Section Setup/Horizon.

In this section, we demonstrate a step-by-step proof of the main claim (Theorem 1 from the paper). It is important to note that Coq's preferred method of proof is a "bottom-up" approach, while most human proofs are rather top-down. In other words, in Coq it is easier to present proofs in the following way: first simple facts are proven, then one can use these simple facts to construct more sophisticated facts, then even more sophisticated ones, and so on until the final goal is reached. Yet, a more practical question for a human would be "where is the goal?"

Therefore, we adopt a more paper-like approach in which we present the proof as a series of implications $C \Rightarrow B, B \Rightarrow A, A \Rightarrow B$. This allows us to begin with the final goal (assuming that some facts are given) and then gradually work our way towards the "truth" of the overall argument pertaining to specific details.

Assume `horizon` defines an upper bound on the termination time of the system. If `horizon` is zero, the system does not necessarily terminate. Note, however, that in either case our proof assumes there to be a finite number of jobs for technical reasons. As the horizon can be chosen to be arbitrarily large (the proof does not depend on its magnitude), e.g. hundreds of even thousands of years, assuming the existence of a finite horizon is not unreasonable for a computing system.

Variable horizon : option instant.
Constant `horizon` : option instant.
Consider any type of tasks with a notion of pWCET...
Content `task` : TaskType;
Content `job` : JobType;
Content `job_task` : JobTask Job Task;

Consider a response time monotonic scheduling algorithm C , where response time monotonic means the following: assuming that all arrival times are fixed, an increase of the execution cost of any job cannot cause a decrease in the response time of any job. Recall that C receives two vectors a vector of arrival times A and a vector of job costs C .

Variable C : JobCosts; **Variable** A : RTMonotonicScheduler horizon C;
Hypothesis $H_{rt_monotonic}$: $rt_monotonic_scheduler$ horizon C;
Hypothesis H_{rt_sched} : rt_sched denotes the probabilistic schedule generated by C for a given system S ;
Let `sched` S := `compute_rt_sched` C (`job_arrival` := `A`, `job` := `C`, `horizon` := `horizon`);
As before, consider four parameters that describe a system under analysis: a sample space Ω , a measure μ , job arrival times a , and job execution costs v .

Variable Ω : countType;
Variable μ : measure T;
Variable a : JobCosts Job;
Variable v : JobCosts Job;
Let us use these parameters to construct a system S .

Let $S := (\Omega, \mu, a, v) := a; a_{arr} := A; C_{job} := C$;
Next, we assume that the above-mentioned pWCET is an axiomatic pWCET. That is, for any job J and any arrival sequence C , there exists a partition of jobs into probabilistic events such that both partition dominance and partition independence are satisfied.

Hypothesis $H_{axiomatic_pWCET}$:
`axiomatic_pWCET` (`job_arr` := `A`, `job` := `C`, `horizon` := `horizon`);
`axiomatic_pWCET` (`job_arr` := `A`, `job` := `C`, `horizon` := `horizon`);

Suppose we use the construction `replace_job_pWCET` presented in problem/rt_monotonic/pWCET_to_pWCET to replace the execution cost of given job J with F_j . Let's denote the resulting system

Variable J_{rep} : Job;
Let $S' := replace_job_pWCET$ J_{rep} S ;
For convenience, let us denote J 's task...
Let $task := job_task$ J_{rep} ;
...and let $task$ denote the measure induced by pWCET.

Let $task :=$
`match` `WCET` `job` `with`
`|` `task` `=>` `product` `WCET` `norming` `task`;
`|` `task` `=>` `task`;
`end` `match`;
`WCET` `task` := `task` `task`;
Next, consider an arbitrary job J of any task and a duration r .

Variables (i, j) : Job (r : duration).
Let `Exc` denote the event that J 's response time exceeds r time units in system S ;
Let `Exc'` := $A \cup B \cup S'$, exceeds `response_time` (i, j) `horizon` S ;
...and let `Exc'` denote the event that J 's response time exceeds r time units in system S' ;
Let `Exc'` := $A \cup B \cup S'$, exceeds `response_time` (i, j) `horizon` S' ;
The remainder of this file serves for the most part to relate the probability of `Exc` with the probability of `Exc'`, namely to establish that $\mu_{job_arr} S \leq \mu_{job_arr} S' \leq \mu_{job_arr} S'$ (Exc) \leq (Exc').

Step 1

Now we are ready to start the proof. First, we do a case analysis on all possible arrival sequences.

Section `Steps`.
Consider a partition `Qpart` of the sample space Ω of S into events corresponding to different arrival sequences.
Section `Step`.
Let `Qpart` := `partition_on_a` μ ;
Here `Qpart` is an (indexed) set of events, where an (i, j) -th event denoted as `Qpart`(i, j) represents a subset of Ω where all arrival sequences is equal to (i, j) .
Therefore focused on individual details may be interesting to notice that the indices of `Qpart`(i, j) are arrival sequences themselves. This detail may be safely skipped over by more casual readers.
As discussed in the paper, one can transfer the partition `Qpart` to the system S' and denote it as `Qpart'`. For further details see the function `partition_transfer_option_partition`.
Let `Qpart'` := `extend_partition` S' `Qpart` S ;
Partitions `Qpart` and `Qpart'` are so similar that one can prove equivalence. For example, two indices (i, j) and (i', j') that are "pickle equivalent" can be shown to be identical. (Pickle equivalence is a very strong notion of equivalence: one can intuitively view it simply as an equality between two elements of similar type, for further details see `utils/leq_utils/pickle`.)
Remark $(L, M, N) \leq (L', M', N')$ iff $(L, M, N) \leq (L', M', N')$.
Now, recall that present the proof in a top-down (\Leftarrow Goal) fashion, starting with the overall horizon.
So, assuming that for any two elements of `partition` $(i, j) \in \text{Qpart}$ and $(i', j') \in \text{Qpart}'$ that are "pickle" equivalent, it holds that $\mu(\text{Exc} \wedge \text{Qpart}(i, j)) \leq \mu(\text{Exc}' \wedge \text{Qpart}'(i', j'))$.
Hypothesis H_{line_indep} : partitioned ;
 $\mu(\text{Exc} \wedge \text{Qpart}(i, j)) \leq \mu(\text{Exc}' \wedge \text{Qpart}'(i', j'))$;
 $\mu(\text{Exc} \wedge \text{Qpart}(i, j)) \leq \mu(\text{Exc}' \wedge \text{Qpart}'(i', j'))$;
...we can show that $\mu_{job_arr} S \leq \mu_{job_arr} S' \leq \mu_{job_arr} S'$ (Exc) \leq (Exc'). In other words, we reduced the lemma statement to the hypothesis statement.
Lemma `transform_line_indep_monotonic_step`;
End `Step`.
Now, we know that if we have the inequality with partitions on arrival sequences $H_{line_indep_partitioned}$, then we are done. But how do we prove such an inequality? In the next step, we prove the required inequality by introducing a new hypothesis from which it can be established. We then continue in this manner until reaching a hypothesis that is easy enough to prove without introducing new hypotheses.

```

...
Step 6
Assume that we are given a list of job arrivals.
Variable (A : Job) - option instant.
For technical reasons, we need to distinguish between jobs costs operating in a probabilistic space with a measure  $\mu_{job\_arr} S$  (Exp S') and the ones operating with the restricted measure  $restrict_{\Omega, S'}(\mu_{job\_arr} S)$  (Exp S').
Recall that a subset of  $\Omega$  that satisfies pickle must take measures into account (to satisfy certain assumptions). Since we restrict  $\mu_{job\_arr} S$  to a subset that satisfies predicate  $(E \wedge S')$  (and a subset of  $\Omega$  that satisfies predicate  $(E' \wedge S')$ ), we have to adjust some of the indices to these new measures.
Let  $Q' := restrict_{\Omega, S'}(\mu_{job\_arr} S)$  (Exp S') := measure  $(Q, S')$ .
Next, we define random variables, which are the same as those introduced earlier, except for the measure  $Q'$  instead of  $\mu$  and  $\mu'$  instead of  $\mu$ .
Let  $Q_j := \text{mkvar } \mu' (E \wedge S')$ ;  

Let  $Q'_j := \text{mkvar } \mu' (E' \wedge S')$ ;  

Similarly, we need to provide a new notion of projection that accounts for the restricted measure.
Variable  $proj$  :=  $\lambda v \lambda i \lambda S' S'' ((E' \wedge S'') \Rightarrow (E \wedge S')) (proj) S'_{rep} \mu$ .
Now, we know that if we have the inequality with partitions on arrival sequences  $H_{line\_indep\_partitioned}$ , then we are done. But how do we prove such an inequality? In the next step, we prove the required inequality by introducing a new hypothesis from which it can be established. We then continue in this manner until reaching a hypothesis that is easy enough to prove without introducing new hypotheses.
Section Steps.
In the second step, we replace the partition over all arrival sequences with an event encoding one arrival sequence.
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into events corresponding to different arrival sequences...
Section Step.
First, let us state the premise of our hypothesis H_line_indep_partitioned.
Again, consider a partition Qpart of the sample space of system  $S$  into
```

AXIOMATIC pWCET IS ADEQUATE

Theorem (paraphrased). Consider a job $J_{i,j}$. Let $\mathcal{R}_{i,j}$ be the pRT of $J_{i,j}$ in the initial system and $\mathcal{R}_{i,j}^\star$ be the pRT of $J_{i,j}$ in a simplified system obtained via pWCET F_j . Then $\mathcal{R}_{i,j} \leq \mathcal{R}_{i,j}^\star$.

Hint:
1. Use axiomatic pWCET to construct a "copy" of the initial system where pETs are replaced with job costs that are

13 steps later ...



If pWCET satisfies our notion of axiomatic pWCET, ...

Hypothesis $H_{axiomatic_pWCET}$:
axiomatic_pWCET ($\mu_of\ S$) ($job_arrival := \mathcal{A}_of\ S$) ($job_cost := \mathcal{C}_of\ S$).

... then the response-time distribution of job j in schedule $sched\ S$ is \leq -bounded by the response-time distribution of job j in schedule $sched\ S'$. That is, $\mathcal{R}_j \leq \mathcal{R}_j'$.

Lemma $prob_rt_monotonic_axiomatic_pWCET_replace_pET$:
 $\mathcal{R}_j \leq \mathcal{R}_j'$.

(This block contains the detailed axiomatic proof steps, including code snippets and mathematical reasoning. It is partially obscured by a large black box in the image.)

Step 1: We start with the initial system and define the job $J_{i,j}$. We introduce the pWCET function F_j and show that it satisfies the axiomatic pWCET hypothesis.

Step 2: We define the simplified system S' and show that it is equivalent to the original system S in terms of the response-time distribution of job j .

Step 3: We use the axiomatic pWCET hypothesis to show that the response-time distribution of job j in S' is \leq -bounded by the response-time distribution of job j in S .

Step 4: We show that the response-time distribution of job j in S' is \leq -bounded by the response-time distribution of job j in S .

Step 5: We show that the response-time distribution of job j in S' is \leq -bounded by the response-time distribution of job j in S .

Step 6: We show that the response-time distribution of job j in S' is \leq -bounded by the response-time distribution of job j in S .

Step 7: We show that the response-time distribution of job j in S' is \leq -bounded by the response-time distribution of job j in S .

Step 8: We show that the response-time distribution of job j in S' is \leq -bounded by the response-time distribution of job j in S .

Step 9: We show that the response-time distribution of job j in S' is \leq -bounded by the response-time distribution of job j in S .

Step 10: We show that the response-time distribution of job j in S' is \leq -bounded by the response-time distribution of job j in S .

Step 11: We show that the response-time distribution of job j in S' is \leq -bounded by the response-time distribution of job j in S .

Step 12: We show that the response-time distribution of job j in S' is \leq -bounded by the response-time distribution of job j in S .

Step 13: We show that the response-time distribution of job j in S' is \leq -bounded by the response-time distribution of job j in S .

FORMAL SPECIFICATION AND PROOFS

Clickable links to Coq specification

→ Each definition, lemma, and proof step is accompanied by a link to the corresponding Coq specification

The LHS and RHS of the inequality can be simplified to $\mathbb{P}[\mathcal{C}_{J_o} > c_0 | \xi \wedge S_l]$ and $\mathbb{P}_f[\widehat{\mathcal{C}}_{J_o} > c_0]$, respectively. Using the fact that $\mathbb{P}[a > b] \leq \mathbb{P}[c > d] \iff \mathbb{P}[a \leq b] \geq \mathbb{P}[c \leq d]$, we transform the inequality to obtain (Ψ):

$$\mathbb{P}[\mathcal{C}_{J_o} \leq c_0 | \xi \wedge S_l] \geq \mathbb{P}_f[\widehat{\mathcal{C}}_{J_o} \leq c_0].$$

Finally, by construction (Def. 10), $\mathbb{P}_f[\widehat{\mathcal{C}}_{J_o} \leq c_0] = F_i(c_0)$. Hence, we end up with $\mathbb{P}[\mathcal{C}_{J_o} \leq c_0 | \xi \wedge S_l] \geq F_i(c_0)$, which follows (Ψ) from partition-dominance (Def. 6). \square



Step 13

In the last step, we exploit the top-level assumption [H_pWCET_bounds_cond_cdf](#) to finish the proof.

Section [Step13](#).

Notice that the following statement is very close to the pWCET guarantee [H_pWCET_bounds_cond_cdf](#).

```
Lemma transformation_is_pRT_monotone_step13 :
  P<μ_of S>{[ e j_rep (<=> c0 | ξf n Sf ]} ≥
  P<μ_tsk>{[ e_pWCET (<=> c0 ]}.
```

Also, note that we did not make any new assumptions in this section; hence, we are done.

End [Step13](#).

FORMAL SPECIFICATION AND PROOFS

Clickable links to Coq specification

- Each definition, lemma, and proof step is accompanied by a link to the corresponding Coq specification
- Links are cumbersome and not clickable in the official IEEE version
- Links are directly clickable in the version provided on the author websites

The LHS and RHS of the inequality can be simplified to $\mathbb{P}[\mathcal{C}_{J_o} > c_0 | \xi \wedge S_l]$ and $\mathbb{P}_f[\widehat{\mathcal{C}}_{J_o} > c_0]$, respectively. Using the fact that $\mathbb{P}[a > b] \leq \mathbb{P}[c > d] \iff \mathbb{P}[a \leq b] \geq \mathbb{P}[c \leq d]$, we transform the inequality to obtain (\heartsuit):

$$\mathbb{P}[\mathcal{C}_{J_o} \leq c_0 | \xi \wedge S_l] \geq \mathbb{P}_f[\widehat{\mathcal{C}}_{J_o} \leq c_0].$$

Finally, by construction (Def. 10), $\mathbb{P}_f[\widehat{\mathcal{C}}_{J_o} \leq c_0] = F_i(c_0)$. Hence, we end up with $\mathbb{P}[\mathcal{C}_{J_o} \leq c_0 | \xi \wedge S_l] \geq F_i(c_0)$, which follows (\heartsuit) from partition-dominance (Def. 6). \square



Step 13

In the last step, we exploit the top-level assumption `H_pWCET_bounds_cond_cdf` to finish the proof.

Section `Step13`.

Notice that the following statement is very close to the pWCET guarantee `H_pWCET_bounds_cond_cdf`.

```
Lemma transformation_is_pRT_monotone_step13 :
  P<μ_of S>{[ e j_rep (<=> c0 | ξf n Sf ]} ≥
  P<μ_tsk>{[ e_pWCET (<=> c0 ]}.
```

Also, note that we did not make any new assumptions in this section; hence, we are done.

End `Step13`.

CONCLUSION

CONCLUSION AND FUTURE WORK

What we did:

- First **fully formal** definitions of pET and pWCET
- **Adequacy property**: formalization of "safe IID upper bound on pET"
- **Prove** that our pWCET proposal is adequate
- All **mechanized** with Coq



coq.inria.fr

CONCLUSION AND FUTURE WORK

What we did:

- First **fully formal** definitions of pET and pWCET
- **Adequacy property**: formalization of "safe IID upper bound on pET"
- **Prove** that our pWCET proposal is adequate
- All **mechanized** with Coq



coq.inria.fr

Are there **better** definitions?

- *Maybe..?*
- *Please propose your preferred definition
... and present an adequacy proof*

CONCLUSION AND FUTURE WORK

What we did:

- First **fully formal** definitions of pET and pWCET
- **Adequacy property**: formalization of "safe IID upper bound on pET"
- **Prove** that our pWCET proposal is adequate
- All **mechanized** with Coq



The Coq Proof Assistant

coq.inria.fr

Are there **better** definitions?

- ▶ *Maybe..?*
- ▶ *Please propose your preferred definition
... and present an adequacy proof*

How to **derive** such axiomatic pWCET?

- ▶ *Are existing methods compatible with it?
(MBPTA? EVT? SPTA?)*
- ▶ *Can compatibility be proven in Coq?*

CONCLUSION AND FUTURE WORK

What we did:

- First **fully formal** definitions of pET and pWCET
- **Adequacy property**: formalization of "safe IID upper bound on pET"
- **Prove** that our pWCET proposal is adequate
- All **mechanized** with Coq



Are there **better** definitions?

- *Maybe..?*
- *Please propose your preferred definition*
... and present an adequacy proof

How to **derive** such axiomatic pWCET?

- *Are existing methods compatible with it?*
(MBPTA? EVT? SPTA?)
- *Can compatibility be proven in Coq?*

Use axiomatic pWCET to build an **analysis**

- *We prove that any sound analysis results in valid bounds. Let's verify one!*

CONCLUSION AND FUTURE WORK

What we did:

- First **fully formal** definitions of pET and pWCET
- **Adequacy property**: formalization of "safe IID upper bound on pET"
- **Prove** that our pWCET proposal is adequate
- All **mechanized** with Coq



coq.inria.fr

Are there **better** definitions?

- *Maybe..?*
- *Please propose your preferred definition*
... and present an adequacy proof

How to **derive** such axiomatic pWCET?

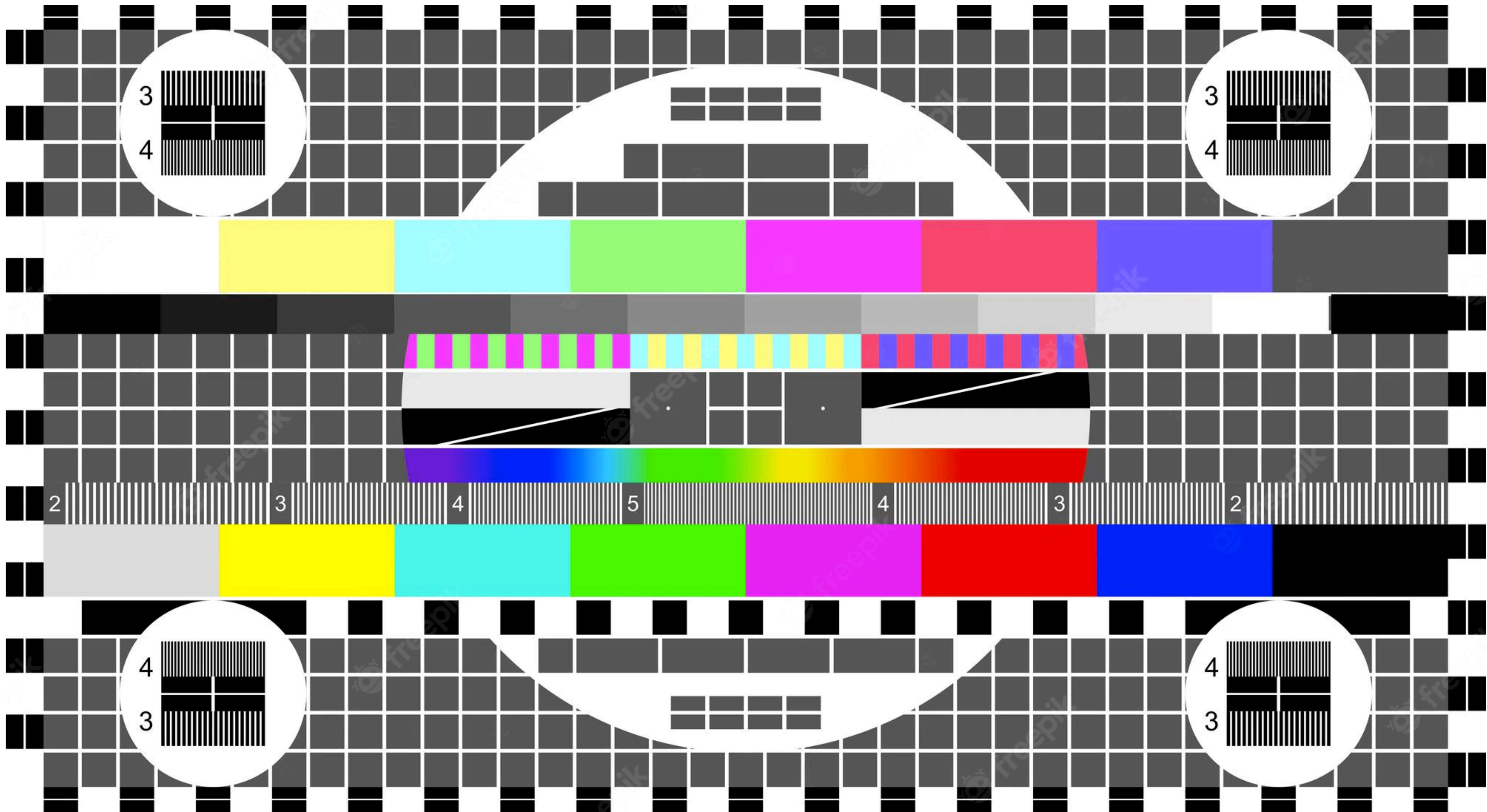
- *Are existing methods compatible with it?*
(MBPTA? EVT? SPTA?)
- *Can compatibility be proven in Coq?*

Still do not like pWCET?

- *Come watch Filip's talk about **pWCET-less***
***Correlation-Tolerant Analysis** on*
December 8th (Session 11 @ 12:35pm)

Use axiomatic pWCET to build an **analysis**

- *We prove that any sound analysis results in*
valid bounds. Let's verify one!



BACKUP SLIDES

WHY AXIOMATIC pWCET?

Theorem (paraphrased). Consider a job $J_{i,j}$. Let $\mathcal{R}_{i,j}$ be the pRT of $J_{i,j}$ in the initial system and $\mathcal{R}_{i,j}^\star$ be the pRT of $J_{i,j}$ in a simplified system obtained via pWCET F_i . Then $\mathcal{R}_{i,j} \leq \mathcal{R}_{i,j}^\star$.

Def. 7 (\clubsuit). A monotonically increasing function $F_i: \mathbb{W} \rightarrow [0, 1]$ with $F_i(0) = 0$ and $\lim_{t \rightarrow \infty} F_i(t) = 1$ is an axiomatic pWCET for a task τ_i if, for every $J \in \tau_i$ and every fixed arrival sequence $\xi \in \Xi$, there exists a partition \mathfrak{S} (Def. 4) such that

- 1) \mathcal{C}_J is partition-independent w.r.t. ξ and \mathfrak{S} (Def. 5), and
- 2) F_i \mathfrak{S} -dominates \mathcal{C}_J w.r.t. ξ (Def. 6).

Hint:

1. Use axiomatic pWCET to construct a "copy" of the initial system, where pETs are replaced with job costs that are, **by construction**, IID and have distribution F_i
2. Prove that pRT $\mathcal{R}_{i,j}^\star$ in the simplified system stochastically dominates the original pRT $\mathcal{R}_{i,j}$

Weakest precondition for which we could find a proof of the adequacy property

TWO TYPES OF pWCET

Dominance pWCET [1]

$$\rightarrow F_i : \mathbb{W} \rightarrow [0,1]$$

\rightarrow Given c , $F_i(c)$ defines a bound on probability of a job of task τ_i to have cost exceeding c

If $F_i(50) = 0.999$, then out of 100,000 jobs, at most 100 jobs are expected to have cost greater than 50

Confidence pWCET [2]

$$\rightarrow F_i : \mathbb{W} \rightarrow [0,1]$$

\rightarrow Given c , $F_i(c)$ defines a bound on probability that WCET of task τ_i does not exceed c

If $F_i(50) = 0.999$, no job is expected to have cost greater than 50 and we are 99.9% confident about it

[1] Davis, Robert I., et al. "Analysis of probabilistic cache related pre-emption delays."

[2] Edgar, Stewart, and Alan Burns. "Statistical analysis of WCET for scheduling."